TPI®
Automotive

Test Process Improvement

(Deutsche Übersetzung)

Version:	1.01
Autor:	Sogeti Deutschland GmbH
Datum:	21.03.2006

0 EINLEITUNG 4
1 TESTSTRATEGIE 10
1.A Teststrategie für einzelne High-Level Tests ... 11
1.B Kombinierte Teststrategie für High-Level Tests ... 14
1.C Kombinierte Strategie für High-Level Tests und Low-Level Tests oder Prüfungsstufen ... 17
1.D Kombinierte Strategie für alle Test- und Prüfungsstufen ... 19
2 EINSATZ DES PHASENMODELLS 21
2.A Planung, Spezifikation, Durchführung ... 21
2.B Vollständiges Phasenmodell Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss ... 25
3 ZEITPUNKT DER BETEILIGUNG 28
3.A Fertigstellung der Testbasis ... 28
3.B Aufstellen der Testbasis ... 29
3.C Aufstellen der Anforderungen ... 29
3.D Beginn des Projekts ... 30
4 KOSTENVORANSCHLAG UND PLANUNG 32
4.A Fundierter Kostenvoranschlag und Planung ... 32
4.B Statistisch fundierter Kostenvoranschlag und Planung ... 34
5 TESTSPEZIFIKATIONSTECHNIKEN 35
5.A Nicht formale Techniken ... 35
5.B Formale Techniken ... 36
5.C Mathematische Methoden ... 37
6 STATISCHE TESTTECHNIKEN 39
6.A Detailüberprüfung der Testbasis ... 39
6.B Checklisten ... 40
7 METRIKEN 41
7.A Projektmetriken (über Produkt) ... 42
7.B Projektmetriken (über Prozess) ... 43
7.C Systemmetriken ... 45
7.D Organisationsmetriken (>1 System) ... 46
8 TESTAUTOMATISIERUNG 47
8.A Einsatz von Testtools ... 48
8.B Beherrschung der Testautomatisierung ... 48
8.C Optimale Testautomatisierung ... 52
9 TESTUMGEBUNG 53
9.A Kontrollierte Testumgebung ... 53
9.B Testen in der geeignetsten Umgebung ... 56
9.C Umgebung auf Abruf ... 57
10 BÜRO- UND LABORAUSSTATTUNG 58
10.A Adäquate und rechtzeitige Einrichtung der Büro- und Laborausstattungen ... 58
11 ENGAGEMENT UND MOTIVATION 60
11.A Zuweisung von Budget und Zeit ... 60
11.B Testen in Projektorganisation integriert ... 61
11.C Testengineering ... 63
12 TESTFUNKTIONEN UND AUSBILDUNGEN 66
12.A Testmanager, Integrator und Tester ... 66
12.B (Formale) methodische, technische und funktionale Unterstützung, Management von Testprozess, Testware und Infrastruktur ... 67
12.C Formale interne Prüfung ... 68
13 REICHWEITE DER METHODIK 70
13.A Projektspezifisch ... 70
13.B Projektspezifisch mit externem Betrachtungsbereich ... 71
13.C Organisationsgenerisch ... 72
13.D Organisationsoptimierend, F&E Aktivitäten ... 73

14 KOMMUNIKATION 75
14.A Interne Kommunikation 75
14.B Projektkommunikation (Abweichungen, Änderungsüberwachung)..... 75
14.C Kommunikation über die Qualität der Testprozesse auf Organisationsebene.... 78
15 BERICHTERSTATTUNG 80
15.A Abweichungen 80
15.B Fortschritt einschließlich Prioritätenzuweisung und Berichterstattung über Zeitaufwand und Testfortschritt 81
15.C Risiken und Empfehlungen anhand von Metriken 82
15.D Empfehlungen haben einen Software Process Improvement Charakter..... 83
16 DOKUMENTATION DER ABWEICHUNGEN 85
16.A Interne Dokumentation der Abweichungen 85
16.B Umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungsmöglichkeiten 86
16.C Dokumentation der Abweichungen wird im gesamten Projekt eingesetzt 87
17 TESTWARE MANAGEMENT 90
17.A Internes Testware Management 90
17.B Externe Verwaltung der Testbasis und des Testobjekts und Zuordenbarkeit von Systemanforderungen zu Testfällen 91
17.C Übertragbare Testware 93
18 TESTPROZESSMANAGEMENT 95
18.A Planung und Durchführung 95
18.B Planung, Durchführung, Überwachung und Anpassung 96
18.C Überwachung und Anpassung in der Organisation 96
19 PRÜFEN 98
19.A Nichtformale Prüfungen 98
19.B Prüfungstechniken 98
19.C Prüfungsstrategie..... 100
20 LOW-LEVEL TESTS 102
20.A Phasenmodell für Low-Level Tests (Planung, Spezifikation und Durchführung) 102
20.B White-Box Techniken..... 105
20.C Low-Level Teststrategie 106
21 INTEGRATIONSTESTS 108
21.A Integration ist als separater und geplanter Prozess identifiziert..... 109
21.B Teststrategie für die Integration..... 110
21.C Standardisierter Ansatz für die Integration 111
22 TESTREIFEMATRIX 113
23 ÜBERBLICK ÜBER DIE ABHÄNGIGKEITEN 114
24 GLOSSAR 117
25 LITERATUR 125

0 EINLEITUNG

In der modernen Kfz-Elektronik und der Softwareentwicklung spielt Testen eine wichtige Rolle im Entwicklungsprozess. Die zunehmende Komplexität dieser elektronischen Baugruppen und der Software fordert die Verbesserung des Entwicklungsprozesses und somit auch des Testprozesses.

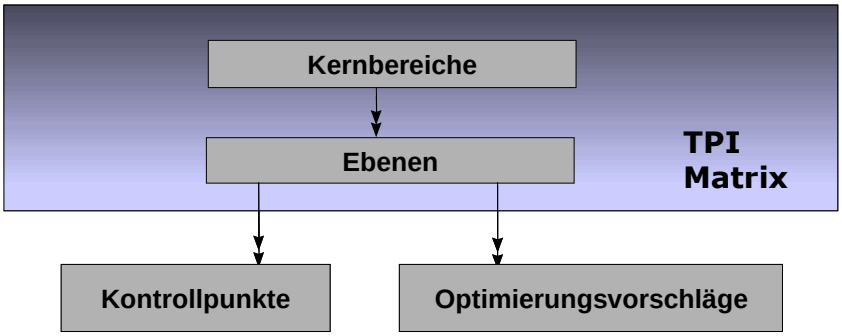
Seit 1998 bietet TPI (Test Process Improvement) [Koomen und Pol, 1999] ein Instrument zur schrittweisen Verbesserung des Testprozesses. Seitdem wurde es in vielen Projekten eingeführt, um dort den Testprozess zu optimieren. Initiiert von den deutschen Automobilherstellern, wurde in den Jahren 2003 und 2004 TPI Automotive entwickelt. Ziel war es ein Modell auf Basis des vorhandenen TPI Modells zu schaffen, welches 'adaptiert' und erweitert wurde, um die Besonderheiten der Automobilindustrie abzudecken.

Das TPI Modell bietet den Rahmen, um die Stärken und Schwächen eines bestehenden Testprozesses innerhalb eines Projekts oder einer Organisation zu bestimmen. Ferner bietet das Modell Unterstützung bei der Formulierung von adäquaten und realisierbaren Optimierungsvorschlägen für den Testprozess in Bezug auf Qualität, Kosten und Zeit an.

Diese Einführung liefert eine kurze Beschreibung des Modells und gibt eine Definition, wie die Bezeichnungen „High-Level Test", „Low-Level Test" und „Integrationstest" in diesem Modell verwendet werden und welche Bedeutung dies für den Gebrauch des Modells hat.

Allgemeine Beschreibung des Modells

Das TPI Modell kann wie folgt dargestellt werden:



Kernbereiche

Im TPI Automotive Modell werden 21 Kernbereiche betrachtet, die alle Aspekte eines strukturierten Testprozesses abdecken. Jeder Kernbereich identifiziert einen typischen Aspekt des Testprozesses. Das TPI Modell bietet Vorschläge zur Verbesserung des Testprozesses in Bezug auf jeden Kernbereich an.

Kernbereich	**Kernbereich**
Teststrategie	Testfunktionen und Training
Einsatz des Phasenmodells	Reichweite der Methodik
Zeitpunkt der Beteiligung	Kommunikation
Kostenvoranschlag und Planung	Berichterstattung
Testspezifikationstechniken	Dokumentation der Abweichungen

Statische Testtechniken	Testware Management
Metriken	Testprozessmanagement
Testautomatisierung	Prüfen
Testumgebung	Low-Level Tests
Büro- und Laborausstattung	Integrationstests
Engagement und Motivation	
…	…

Ebenen

Jeder Kernbereich wird in unterschiedliche Ebenen der Reife unterteilt. Ein Kernbereich kann in bis zu vier Ebenen unterteilt sein, die mit den Buchstaben A bis D bezeichnet sind. Jede nächsthöhere Ebene ist in zeitlicher, finanzieller und/oder qualitativer Hinsicht reifer als die jeweils vorhergehende Ebene.

Durch die Einteilung in Ebenen kann die derzeitige Reife eines Testprozesses klarer festgestellt werden und dadurch können bessere Ziele für eine schrittweise Optimierung vorgeschlagen werden.

Kontrollpunkte

Zur objektiven Bestimmung der Ebene, auf der sich ein Testprozess befindet, verfügt das Modell über ein Messinstrument, die so genannten Kontrollpunkte. Jede Ebene hat einige Kontrollpunkte. Ein Testprozess muss diese Punkte erfüllen, um in diese Ebene eingeteilt zu werden. Die Kontrollpunkte sind kumulativ, d.h. um für Ebene B in Betracht zu kommen, muss der Testprozess sowohl den Kontrollpunkten von Ebene B als auch denen von Ebene A entsprechen.

Testreifematrix (TPI Matrix)

Die Kernbereiche und ihre Reifegradebenen werden in der Testreifematrix (TPI Matrix) kombiniert.
Die Verteilung der Reifegradsebenen basiert auf Prioritäten und Abhängigkeiten zwischen den Ebenen verschiedener Kernbereiche.
Um zum Beispiel mit Testautomatisierung zu beginnen, müssen zunächst Testfälle existieren. Diese Testfälle lassen sich durch Testspezifikationstechniken ableiten. Daher hat Ebene A des Kernbereichs „Testspezifikationstechniken" eine höhere Priorität als Ebene A des Kernbereichs „Testautomatisierung". Deshalb wird Ebene A des Kernbereichs „Testspezifikationstechniken" auch weiter links in der Matrix platziert als Ebene A des Kernbereichs „Testautomatisierung".

Ebene A des Kernbereichs „Metriken" setzt folgendes voraus:
Ebene B des Kernbereichs „Testprozessmanagement"
Ebene A des Kernbereichs „Dokumentation der Abweichungen"
Ebene B des Kernbereichs „Berichterstattung"
Ebene B des Kernbereichs „Engagement und Motivation"

Ebene A des Kernbereichs „Metriken" befindet sich in der Matrix daher ebenso weit rechts wie Ebene B des Kernbereichs „Engagement und Motivation" (Alle Abhängigkeiten zwischen den verschiedenen Ebenen der Kernbereiche werden in Kapitel 23 dargestellt).

Bevor Optimierungsvorschläge formuliert werden können, ist die jeweilige Ebene für jeden einzelnen Kernbereich zu bestimmen. Es wird eine Analyse des Testprozesses durchgeführt, um zu bestimmen, ob die Kontrollpunkte der Kernbereiche erfüllt sind.

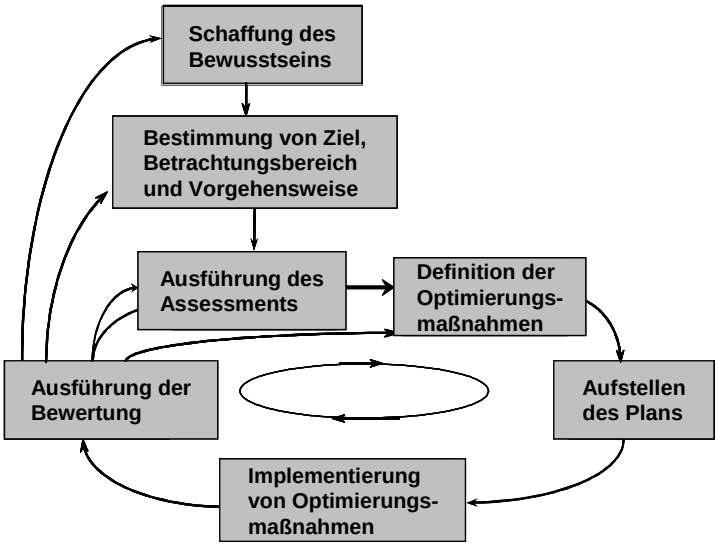
Wenn alle Kontrollpunkte erfüllt sind, ist die jeweilige Ebene für den betreffenden Kernbereich erreicht.
Die Matrix wird nach dieser Betrachtung des Testprozesses ausgefüllt und bietet so allen Beteiligten eine übersichtliche Vorstellung davon, auf welcher Ebene sich der Testprozess befindet. Anhand der Matrix können anschließend gezielte Überlegungen stattfinden, welche Reifegrade für welche Kernbereiche erreicht werden sollen.
Indikationen, welche Maßnahmen man ergreifen sollte, um den Testprozess auf die gewünschten Ebenen zu bringen, werden in Form von Optimierungsvorschlägen formuliert.

Optimierungsvorschläge

Das TPI Modell formuliert die zu ergreifenden Maßnahmen zur Optimierung des Testprozesses auf den gewünschten Ebenen. Diese Maßnahmen sind in Form von allgemeinen Optimierungsvorschlägen formuliert.
Diese Optimierungsvorschläge können als Grundlage herangezogen werden, um konkrete, spezifische Verbesserungsmaßnahmen (bezogen auf den jeweiligen analysierten Prozess) auszuarbeiten.
Die Implementierung dieser Verbesserungsmaßnahmen sollte die Organisation für die betroffenen Kernbereiche auf die jeweils gewünschte Ebene bringen.

Anwendung des TPI Automotive Modells

Jeder Änderungsprozess besteht im Allgemeinen aus den gleichen Arbeitsschritten. Auf der Grundlage von Zielsetzungen werden Änderungen ausgeführt, um vom gegenwärtigen Zustand zu der gewünschten Situation zu gelangen. Die Optimierung des Testprozesses unterscheidet sich hierbei nicht wesentlich von jedem beliebigen anderen Änderungsprozess.
Das folgende Diagramm stellt die Aktivitäten eines Änderungsprozesses dar. Diese Aktivitäten werden beschrieben, wobei auf die Elemente besonders eingegangen wird, bei denen das TPI Automotive Modell eine spezifische Rolle spielen kann:



Schaffung des Bewusstseins

Der Grund für eine Optimierung des Testprozesses liegt meistens im Erkennen gewisser Probleme im Bereich des Testens. Für diese Probleme will man Lösungen finden und eine Optimierung des Testprozesses wird als eine solche Lösung angesehen.

Dieses Bewusstsein beinhaltet, dass alle Beteiligten untereinander über die wichtigsten Punkte des Änderungsprozesses einig werden und ihre Unterstützung zusichern. Das Engagement darf nicht nur zu Beginn des Änderungsprozesses vorhanden sein, sondern muss sich durch alle Phasen des Prozesses ziehen. Dies erfordert regelmäßige Informationsverteilung und Meetings.

Bestimmung von Ziel, Betrachtungsbereich und Vorgehensweise

Bei dieser Aktivität werden insbesondere Ziel, Betrachtungsbereich und Vorgehensweise des Änderungsprozesses bestimmt.
Sollte das Testen kostengünstiger, schneller oder besser werden?
Welche Teststufen und welche Testprojekte werden betrachtet?
Wie viel Zeit steht für die Verbesserungsmaßnahmen zur Verfügung?
Welche Kosten werden veranschlagt bzw. welches Budget steht zur Verfügung?

Ausführung der Analyse

Bei der Analyse werden die wichtigen und weniger wichtigen Aspekte der aktuellen Situation aufgenommen.
Der Einsatz des TPI Automotive Modells ist ein wesentlicher Bestandteil der Analyse, da das Modell einen Rahmen bietet, anhand dessen die genannten wichtigen und schwachen Aspekte des untersuchten Testprozesses aufgenommen werden können.
Auf der Grundlage der gesammelten Informationen werden für jeden Kernbereich des TPI Automotiv Modells gesondert anhand der Kontrollpunkte die Ebenen bestimmt und es wird festgelegt, bei welchen Kontrollpunkten ganz, teilweise oder nicht entsprochen wurde. Die TPI Matrix wird eingesetzt, um die Situation des Testprozesses im Überblick darzustellen.
Die Matrix zeigt die Stärken und Schwächen des Testprozesses eindeutig auf, indem die jeweils erreichten Ebenen pro Kernbereich in der Matrix eingefärbt werden.

Definition der Optimierungsmaßnahmen

Auf der Grundlage der Optimierungszielsetzungen und dem Ergebnis der Analyse werden die Optimierungsmaßnahmen ermittelt. Die Maßnahmen werden so definiert, dass eine allmähliche und schrittweise Optimierung durchgeführt werden kann.

Das TPI Automotive Modell hilft bei der Bestimmung der Optimierungsmaßnahmen. Die Ebenen der Kernbereiche und die TPI Matrix bieten eine Reihe von Möglichkeiten, um Optimierungsschritte festzulegen. Abhängig von den Zielen, dem Betrachtungsbereich, der Durchlaufzeit und den Ergebnissen der Analyse kann entschieden werden, ob man ein oder mehrere Kernbereich(e) optimiert. Bei jedem einzelnen ausgewählten Kernbereich kann man sich ferner dafür entscheiden, in die nächste Ebene oder in Sonderfällen sogar in eine darauf folgende nächsthöhere Ebene zu wechseln.
Außerdem liefert das TPI Automotive Modell eine Vielzahl an Optimierungsvorschlägen, die dabei helfen, die gewünschten höheren Ebenen zu erreichen.

Aufstellen des Plans

Ein Plan wird formuliert, um Verbindlichkeit und Engagement aller Beteiligten für die zu implementierenden Aktivitäten zu erhalten. Der Plan beinhaltet Zielsetzungen und Planungsaktivitäten, um die Zielsetzungen zu erreichen. Der Plan bezieht sich sowohl auf die inhaltlichen Aktivitäten zur Optimierung des Testprozesses als auch auf die Aktivitäten, die erforderlich sind, um den Änderungsprozess in die richtige Bahn zu lenken.

Implementierung von Optimierungsmaßnahmen

Der Plan wird ausgeführt. Die Widerstände, die zweifelsohne vorhanden sind, müssen offen gelegt und besprochen werden. Bei den jeweiligen Maßnahmen muss gemessen werden, ob und inwieweit diese ausgeführt wurden. Hilfsmittel sind dabei so genannte „self analysis" (Selbsteinschätzungen), wobei das TPI Automotive Modell eingesetzt wird, um schnell den Zustand des Testprozesses festzustellen. Dabei untersuchen die Beteiligten anhand des TPI Automotive Modells ihren eigenen Testprozess. Ein weiteres wesentliches Element in dieser Phase ist die Konsolidierung. Es ist zu vermeiden, dass die implementierten Optimierungsmaßnahmen einen einmaligen oder vorübergehenden Charakter haben. Die veränderte Arbeitsweise muss weiterhin angewendet werden.

Ausführung der Bewertung

Inwieweit haben die implementierten Maßnahmen das gewünschte Ziel erbracht? In dieser Phase wird festgestellt, in welchem Umfang die Maßnahmen erfolgreich implementiert worden sind und inwiefern die ursprünglichen Ziele erreicht wurden.

Umfang von High-Level , Low-Level und Integrationstests

Im TPI Automotive Modell werden drei unterschiedliche Teststufen unterschieden:

- High-Level Tests
- Low-Level Tests
- Integrationstests

Bei *High-Level Tests* werden ganze, vollständige Produkte getestet. Ein ganzes, vollständiges Produkt bezeichnet ein Produkt so wie es bei der Beauftragung für dieses Produkt definiert wurde. Der Zweck von High-Level Tests ist aufzuzeigen, in welchem Umfang die Anforderungen im System implementiert wurden.

Low-Level Test ist der Prozess des Testens individueller Komponenten, eine nach der anderen oder in Kombination. Um eine Komponente korrekt zu testen, muss der Tester üblicherweise Stubs und Treiber einsetzen oder diese durch echte Komponenten ersetzen.

Integrationstest ist der Prozess, Fehler in Schnittstellen bzw. im Zusammenspiel verschiedener integrierter Komponenten aufzuzeigen. Im Gegensatz zum Low-Level Test liegt der Schwerpunkt auf dem Zusammenspiel der Module, Komponenten etc. Gegenstand und Umfang der Integrationstests verändern sich im Laufe des Entwicklungsprozesses.

In den ersten Phasen des Entwicklungsprozesses liegt der Schwerpunkt beim Integrationstest von Komponenten auf der niedrigsten Ebene.
Das Testen unterscheidet sich anfangs nicht so sehr vom Low-Level Test. Gegen Ende des Entwicklungsprozesses versteht man unter Integrationstests die Kombination kompletter Subsysteme, wobei der Charakter des Integrationstests eher dem von High-Level Tests entspricht.

Dennoch findet für diese beiden Gegensätze derselbe Kernbereich (21) Anwendung, da die Prozesse des Integrationstests dieselben sind.
Obwohl das Modell diese drei Testebenen unterscheidet, bedeutet dies nicht, dass alle drei Ebenen für jede Analyse Untersuchungsgegenstand sein müssen.
Die Anwendung des Modells in einer Situation, bei der High-Level Tests Gegenstand sind, ist offensichtlich. Die Kernbereiche 1 bis 19 lassen sich auf diese Situation anwenden.

Wenn auch Low-Level Tests und Integrationstests Gegenstand der Analyse sind, so sind auch die Kernbereiche 20 und 21 anwendbar.
In Situationen, wo zwei oder mehrere Teststufen mehr als eine Organisation oder Organisationsbereiche umspannen, gibt es zwei Möglichkeiten, das Modell einzusetzen.
Beim ersten Ansatz werden die Teststufen als ein gemeinsamer Prozess analysiert und die Ergebnisse werden in einer TPI Matrix eingetragen.
Beim zweiten Ansatz werden die einzelnen Teststufen individuell analysiert und die Ergebnisse werden in separaten TPI Matrizen je Teststufe präsentiert.
Für High-Level und Low-Level Tests bedeutet dies, dass die Kernbereiche 1 bis 19 und für Integrationstests zusätzlich der Kernbereich 21 anzuwenden sind.
In der Praxis jedoch werden Low-Level Tests meist berücksichtigt, sobald eine oder mehrere der übrigen Teststufen analysiert werden.

1 TESTSTRATEGIE

Einer der wichtigsten Kernbereiche ist die Teststrategie, deren Ziel es ist:

So früh und so kostengünstig wie möglich die wichtigsten Fehler zu erkennen.

Die Formulierung „wichtigsten" bezieht sich auf die Risiken für die Organisation, wenn sich das Produkt als qualitativ unzureichend herausstellt. In der Teststrategie wird bestimmt, welche Anforderungen und Risiken mit welchen Tests abgedeckt werden. Je besser jede Teststufe die eigene Strategie bestimmt und je besser diese verschiedenen Teststrategien einander angepasst werden, desto höher wird die Qualität der gesamten Teststrategie.

Die Optimierung des Testprozesses beginnt in erster Linie bei den High-Level Tests. Deshalb durchläuft ein Testprozess für diesen Kernbereich folgende Reifegradebenen: In der ersten Ebene wird mit einer Teststrategie für einen einzelnen High-Level Test begonnen, in der nächsten Ebene findet die Koordination der Strategie zwischen den verschiedenen High-Level Tests statt, bis auf der höchsten Ebene eine Gesamtstrategieabstimmung zwischen sämtlichen Test- und Prüfungsstufen erfolgt.

Ein Merkmal der Startebene ist, dass der Test nur über die Ressourcen und die Zeit kontrolliert wird. Meistens wird nur eine einzige Testspezifikationstechnik angewandt und nur die Funktionalität getestet. Ferner findet zwischen den verschiedenen Teststufen keine Abstimmung im Zusammenhang mit den zu überprüfenden Qualitätsmerkmalen, dem Betrachtungsbereich des Tests usw. statt.

Die Teststrategie basiert auf einer Risikoeinschätzung (Risiko basierte Teststrategie). Diese Risikoeinschätzung bietet die Möglichkeit, das ideale Gleichgewicht zwischen der gewünschten Qualität und dem erforderlichen zeitlichen und finanziellen Umfang zu finden. Komponenten mit einem hohen spezifischen Risiko werden in vollem Umfang Gegenstand des Tests sein. Die Risikoabschätzung in Verbindung mit der Größe des Teils bestimmt den erforderlichen zeitlichen und finanziellen Aufwand, um dieses Teil zu testen.
Essentielle Schritte auf dem Weg zu einer risikobasierten Teststrategie sind:

- Festlegen der Qualitätskriterien;
  In Abstimmung mit den betroffenen Parteien werden die Qualitätskriterien festgelegt, auf welche der Test sich fokussieren soll. Während des Testprozesses werden die Testergebnisse an den Auftraggeber berichtet.
- Festlegen der relativen Bedeutung der Qualitätskriterien;
  Basierend auf den Ergebnissen des vorhergehenden Schritts wird aufgezeigt, wie der Testaufwand auf die gewählten Qualitätskriterien verteilt werden sollte. Der Ausgangspunkt ist eine Gleichverteilung des Testaufwands auf alle Qualitätskriterien
- Zerlegen in Systemkomponenten;
  In diesem Schritt wird das System auf Basis der logischen Systemarchitektur in einzelne Systemkomponenten zerlegt. Wenn möglich, sollte das System so zerlegt werden, dass jede Systemkomponente einem einzigen Lieferanten zugeordnet werden kann. Wird von dieser Unterteilung abgewichen, muss dies ausdrücklich begründet sein. Ferner wird oft die Komponente „Gesamtsystem" berücksichtigt, um die relative Bedeutung des Tests des Gesamtsystems darzustellen. Die Idee dabei ist, Risiken zu berücksichtigen, die durch separate Tests aller Einzelkomponenten nicht abgedeckt werden.

- Festlegen der relativen Bedeutung der Systemkomponenten;
  Basierend auf den Ergebnissen des vorigen Schritts wird nun festgelegt, wie der Testaufwand auf die einzelnen Systemkomponenten verteilt wird. Der Ausgangspunkt ist auch hier wieder eine Gleichverteilung des Aufwands auf alle Systemkomponenten. Als nächstes wird ermittelt, welche Qualitätskriterien für jede einzelne Systemkomponente anzuwenden sind und wie gründlich diese in Bezug auf ihre zugewiesene Bedeutung zu testen sind. Wenn die Zerlegung zu einer 1:1 Beziehung zwischen Systemkomponenten und Lieferanten geführt hat, können den Lieferanten konkrete Eingangs- und/oder Annahmekriterien mitgeteilt werden.
- Festlegen von Messtechniken und Testspezifikationstechniken;
  Als letzter Schritt innerhalb der Teststrategie werden Metriken und vor allem Testspezifikationstechniken ausgewählt, welche für die zuvor festgelegten Qualitätsmerkmale und Systemkomponenten am besten geeignet erscheinen.

Während der Bestimmung der Teststrategie sind folgende Punkte bei der Risikobetrachtung besonders zu berücksichtigen:

- Die Funktionalität der Systemkomponenten.
- Die Komplexität der Systemkomponenten und ihrer Schnittstellen.
- Die von den Lieferanten zur Verfügung gestellten Qualitätsnachweise (in Form von Testfällen, Testberichten, Zertifikaten, etablierte Anwenderbasis, usw.) oder die in den Eingangskriterien festgelegten geforderten Qualitätsnachweise.
- Die Transparenz der Systemkomponenten. Transparenz bedeutet, mit wie wenig Aufwand man ermitteln kann, ob ein Systemteil funktioniert und deshalb als Indikator dafür steht, wie leicht es ist, die Ursache für Ausfälle herauszufinden. Transparenz kann hergestellt werden z.B. durch ursprüngliche Spezifikationen, die der Lieferant von Systemkomponenten bereitstellt, bzw. durch bereitgestellten Quellcode oder durch so genannte Testfenster in der Software, die Zwischenergebnisse ausgeben.
- Die Häufigkeit der Verwendung der Systemkomponenten.
- Der mögliche Schaden, wenn eine spezifische Komponente der Software versagt.
- Die Verfügbarkeit von (Test-) Ressourcen (sowohl Personal als auch (Test-) Infrastruktur).
- Der Ausgangspunkt des Risikoanalyseteils der Teststrategie sind die funktionalen und nichtfunktionalen Anforderungen, mögliche Standards oder Vorschriften und andere Beschränkungen. Beispiele könnten Performanceanforderungen, Sicherheitsvorschriften (wie z. B. ISO 61508) und Speicherbeschränkungen (ISO: Ressourcenverwendung) sein.

Sicherheitsrelevante Systeme oder Systemkomponenten nehmen eine ganz besondere Rolle bei der Diskussion von Risiken ein. Obwohl man auch hier zwischen großem und geringem Einfluss bezüglich der Sicherheit unterscheiden kann, ist es völlig indiskutabel, solche Bereiche nicht zu testen. Die Sorgfalt des Testens und die eingesetzten Testspezifikationstechniken sind die Instrumente, die sich durch die Risikoeinschätzung beeinflussen lassen (Obwohl ISO 61508 hier die Möglichkeiten einschränkt).

1.A Teststrategie für einzelne High-Level Tests

Beschreibung

Der Auftraggeber eines Tests erwartet vom freizugebenden System bestimmte Qualitäten, die von Fall zu Fall sehr unterschiedlich sind. Es ist von größter Bedeutung, dass hierüber Vereinbarungen mit dem Auftraggeber getroffen werden und dass -

abhängig von den Wünschen des Auftraggebers - diese in eine entsprechende Testvorgehensweise umgesetzt werden.
Besprechungen mit dem Auftraggeber über eine Teststrategie stellen sich häufig als einen Erkenntnisprozess heraus. Anstatt nur Entscheidungen in Hinsicht auf Geld und Zeit zu treffen, stehen plötzlich auch ganz andere Auswahlmöglichkeiten zur Diskussion. Durch eine Detaillierung wird der Testprozess zudem kontrollierbarer. Die Informationen, die man während der Ausarbeitung einer risikobasierten Teststrategie gewinnt sind sehr wertvoll, wenn es darum geht, zwischen der geforderten Qualität, Zeit und Geld abzuwägen. Wenn ein Auftraggeber will, dass das Testen kostengünstiger erfolgt, kann der Testmanager ihn vor die Entscheidung stellen, einen bestimmten Test wegfallen zu lassen oder einen anderen Test in abgeschwächter Form, also weniger intensiv, durchzuführen. Dies bietet dem Auftraggeber später die Möglichkeit, sein eigenes Budget zu verteidigen.

Beispiel 1:

Ein Automobilproduzent will eine neue Version einer bestehenden Software implementieren. Für diese neue Version ist ein Abnahmetest geplant und hierfür wird eine risikobasierte Teststrategie bestimmt. Bei der Formulierung dieser Strategie wird deutlich, dass die neue Version keine neue Software ist, sondern eine neu parametrisierte Version der bestehenden Software. Diese Information führt zu der Schlussfolgerung, dass Risiken aus Änderung des Programmcodes nicht existieren und somit keine Low-Level Tests erforderlich sind. In diesem Fall kann sich die Risikoabschätzung darauf beschränken, dahingehend zu analysieren, welche Anforderungen in Bezug zu den Parametereinstellungen stehen und für welche Anforderungen ein Regressionstest erforderlich ist. Diese Analyse spart viel Zeit und Mühe und ermöglicht es, sich auf die Risiken zu konzentrieren, die man tatsächlich noch abdecken muss.

Kontrollpunkte

1.A.1 Es findet eine fundierte Risikoüberlegung statt. Typische Risikokategorien, die es zu überprüfen gilt, sind: technische Risiken (eine FMEA Einflussanalyse kann hierzu als Grundlage dienen), organisatorische Risiken, die mit der Entwicklung und dem Test in Zusammenhang stehen, der operative Einsatz des Produkts, politische und vertragliche Risiken sowie Haftungsfragen.

1.A.2 Diese Überlegung beinhaltet zumindest die folgenden Aspekte:

- Regressionstests nicht modifizierter Komponenten der Software sind ein Teil dieser Strategie, sofern das Testobjekt ein Update oder ein neues Release bestehender Software darstellt.
- Software wird oft parametrisiert, z. B. auf Grund länderspezifischer Gesetze. In solchen Fällen sollte es ein Teil der Teststrategie sein, festzulegen ob und wie die unterschiedlichen Parametrisierungsmöglichkeiten basierend auf den geschätzten Risiken gestestet werden können.
- Die zu testende Software kann entweder aus kommerzieller Standardsoftware (sog. COTS – commercial off the shelf), wiederverwendeter Software, dem Programmkern bzw. komplett neuer Programmierung bestehen oder eine Kombination aus den genannten Beispielen sein. Die Teststrategie muss die unterschiedlichen Risikoprofile von Standardsoftware, wiederverwendeter Software, dem Programmkern, komplett neuer Programmierung oder eine Kombination davon berücksichtigen.
- Risiken werden berücksichtigt, die in Abhängigkeiten der Produkte bezogen auf deren Hardware- und Softwarebasiskonfiguration beruhen, wie z. B. geforderte Rückwärtskompatibilität oder Hardwaresprünge.

1.A.3 Alle Interessengruppen des Produktes sind in den Prozess der Bestimmung der Teststrategie involviert. Zumindest müssen alle Betroffenen (vor allem der Produktverantwortliche) eingeladen werden, die vorgeschlagene Teststrategie und ihren gegenwärtigen Status zu prüfen.

1.A.4. Es erfolgt eine Differenzierung in der Tiefe der Tests, abhängig von den erkannten Risiken und - falls vorhanden - den Eingangs- und Akzeptanzkriterien. Es werden weder alle Teilsysteme, Varianten und Versionen gleich intensiv getestet, noch wird jedes Qualitätsmerkmal (gleich intensiv) getestet. Wenn die Funktionalität inkrementell bereitgestellt wird, (z.B. A-Muster, B-Muster, etc.), muss für jedes Inkrement eine klare Entscheidung getroffen werden, was getestet werden soll und welche Regressionstests durchgeführt werden müssen.

1.A.5 Es werden eine oder mehrere Testspezifikationstechniken angewandt, abhängig von der jeweilig gewünschten Tiefe des Tests.

1.A.6 Für erneute Tests findet ebenfalls eine (einfache) Strategiebestimmung statt, wobei eine fundierte Auswahl zwischen den Möglichkeiten „nur Lösungen testen" und „vollständigem Regressionstest" getroffen wird.

Abhängigkeiten

- Testspezifikationstechniken, Ebene A, nicht formale Techniken
  Testspezifikationstechniken sind erforderlich, um die Wahl zwischen einem leichteren oder einem schwierigeren Test zu konkretisieren.
- Engagement und Motivation, Ebene A, Zuweisung von Budget und Zeit
  Mit dem Auftraggeber des Tests muss die Strategie gesprochen werden können, da diese stark im Zusammenhang mit den erforderlichen zeitlichen und finanziellen Mitteln steht.

Optimierungsvorschläge

Beziehen Sie die verschiedenen Beteiligten, wie Auftraggeber, Systemadministratoren und Projektmanager bei der Bestimmung der Teststrategie ein.

Schaffen Sie „Bewusstsein", indem Sie die Risiken bei der heutigen Arbeitsweise angeben, oder schlagen Sie vor, wie das Testen kostengünstiger bzw. schneller verlaufen kann.

Wenn nur eine Testspezifikationstechnik vorliegt, so versuchen Sie, anhand einfacher Variationen die Möglichkeit zu schaffen, mehr bzw. weniger Tiefe anzusetzen. Als Beispiel von mehr oder weniger Tiefe kann das Testen oder Nichttesten von Grenzwerten genannt werden. Grenzwerte sind Werte, die kurz vor, auf oder kurz hinter der Grenze eines Bereichs liegen. Bei der Bedingung „A>=10" gelten 9, 10 und 11 bei ganzen Zahlen als Grenzwerte.

Stellen Sie für erneute Tests eine Arbeitsweise auf, bei der jeweils die Abwägung zwischen vollständigem Regressionstesten, eingeschränktem Regressionstesten (je Fehler/Funktion/Teilsystem) oder sogar kein Regressionstest erfolgen kann (und dies schriftlich festgehalten wird). Für sicherheitsrelevante Software ist ein vollständiger Nachtest meist obligatorisch.

Unterscheiden Sie zwischen den verschiedenen Teilsystemen und Qualitätsmerkmalen und versuchen Sie, jedem Teilsystem und jedem Qualitätsmerkmal eine relative Bedeutung zuzuweisen. Setzen Sie diese Bedeutung in einen umfangreichen oder weniger umfangreichen Test um.

Um das Risiko einer verspäteten Lieferung bestimmter Systemkomponenten zu minimieren, sollte man über Rückfallszenarien nachdenken. Es kann helfen, eine Systemkomponente durch eine vorhergehende Version mit mehr oder weniger derselben Funktionalität zu ersetzen. Das Risiko der Regression (eine bereits getestete Systemkomponente funktioniert nicht mehr nachdem eine neue Version einer anderen Systemkomponente eingesetzt wird) sollte in diesem Fall besonders bedacht werden.

Im Falle inkrementeller Entwicklung wird die Festlegung und Pflege eines Regressionstests viel Zeit sparen. Der Regressionstest kann schrittweise für jedes Inkrement aufgebaut werden. In diesem Fall sollte man über die Verwendung von Testtools nachdenken.

Führen Sie schließlich eine vollständige Strategiebestimmung durch.

Um parametrisierte Software zu testen, gibt es mehrere Möglichkeiten:

- Testen der Software unter Verwendung unterschiedlicher Parametersätze (arbeitet die Software mit den eingestellten Parametern korrekt?).
- Überprüfen oder Testen, ob die unterschiedlichen eingesetzten Parametersätze korrekt sind (sind die Parameter für jede Situation korrekt eingestellt?).
- Wenn die Anzahl der möglichen Parametersätze zu hoch ist (aufgrund einer großen Variantenanzahl oder großer regulatorischer Unterschiede zwischen verschiedenen Ländern), ist eine Risikoabschätzung notwendig, um herauszufinden, welche Parametersätze während des Tests (teilweise) außer Acht gelassen werden können.

Kommerzielle Standardsoftware hat im Vergleich zu individuell erstellter Software ein anderes Risikoprofil. Als Faustregel sind die funktionalen Risiken tendenziell geringer, da andere Anwender der Software wahrscheinlich bereits viele der bestehenden funktionalen Risiken gefunden haben. Auf der anderen Seite sind die Risiken hinsichtlich der Eignung für die Organisation höher, da sie so konzipiert wurde, dass sie für alle möglichen Organisationen passen soll („One size fits all").
Identifizieren Sie den Auftraggeber des Tests und lassen Sie ihn seine Anforderungen an den Test klar darlegen.

1.B Kombinierte Teststrategie für High-Level Tests

Beschreibung

Die Abstimmung der Teststrategien der verschiedenen High-Level Tests verhindert, dass Tests doppelt ausgeführt werden oder dass „Löcher" zwischen den Tests auftreten. Außerdem ist man besser in der Lage, die Tests zum richtigen Zeitpunkt durchzuführen, d.h., wenn (bei einer übereinstimmenden Testintensität) die Testkosten zzgl. der Korrektur- und Regressionstestkosten am niedrigsten sind. Diese Abstimmung setzt voraus, dass bekannt ist, welche High-Level Tests geplant sind und was und mit welcher Intensität in jedem High-Level Test getestet wird.
Die Qualität der Kommunikation zwischen Auftraggeber und Lieferant ist hierbei überaus wichtig. Beide müssen ihr gemeinsames Interesse daran zeigen, Informationen über die vom Lieferanten durchgeführten Tests zu fordern bzw. zu liefern.
Der Auftraggeber muss die richtigen Informationen einfordern und der Lieferant sollte einen klaren Einblick liefern was er zu testen plant und was nicht.

Kontrollpunkte

1.B.1 Es erfolgt eine Abstimmung zwischen den verschiedenen High-Level Tests – vor allem dem System-, Abnahme- und Produktionsfreigabetest oder Tests auf Lieferanten- und Auftraggeberseite - auf dem Gebiet der Teststrategie (Risiken, Qualitätsmerkmale, Betrachtungsbereich des Tests und der Planung).

1.B.2 Das Ergebnis der Abstimmung ist eine koordinierte Strategie, die schriftlich festgelegt und während des gesamten Testprozesses überwacht wird.

1.B.3 Jeder High-Level Test bestimmt auf der Grundlage der koordinierten Strategie die eigene Teststrategie, wie bei Ebene A beschrieben.

1.B.4 Abweichungen von der koordinierten Strategie werden gemeldet. Eine fundierte Anpassung der koordinierten Strategie wird auf Basis der aus dieser Abweichung resultierenden Risiken vorgenommen. Im Falle inkrementeller Lieferungen wird für jedes einzelne Inkrement die Strategie validiert.

1.B.5 Für Nachtests findet eine Koordination zwischen den unterschiedlichen Teststufen statt. Sofern die verschiedenen Teststufen die Organisationsgrenzen überschreiten, fällen Auftraggeber und Lieferant auf Basis der Eingangs- und Ausgangskriterien klare Entscheidungen über Nachtests auf beiden Seiten.

Abhängigkeiten

- Einsatz des Phasenmodells, Ebene A, Planung, Spezifikation, Durchführung
  Zur Abstimmung müssen im Vorfeld des Testens Vereinbarungen getroffen werden, was, wie und wann zu testen ist (Planungsphase). Die durchzuführenden Aktivitäten müssen dokumentiert werden. Dies erfordert einen transparenten Prozess, für den der Einsatz des Phasenmodells eine Voraussetzung ist.
- Testspezifikationstechniken, Ebene B, formale Techniken
  Die Koordination der unterschiedlichen Tests erfordert einen tieferen Einblick in die Qualität und die Tiefe jedes einzelnen Tests.
  Das heißt, dass formalere Testspezifikationstechniken eingesetzt werden müssen.
- Engagement und Motivation, Ebene B, Testen ist in die Projektorganisation eingebettet. Um die Koordination zu etablieren, ist das Engagement des Projektmanagements erforderlich. Daher muss das Management bemüht sein, einen Einblick in die Qualität und Intensität des Testens zu erhalten.
- Kommunikation, Ebene B, Projektkommunikation, Analyseforum, Änderungsüberwachung
  Eine Abstimmung zwischen den Teststufen und dem übrigen Projekt hat während des gesamten Testprozesses zu erfolgen.
- Testprozessmanagement, Ebene B, Planung, Durchführung, Überwachung und Anpassung
  Einsicht in die Qualität eines jeden Tests bedeutet, dass man sich nicht nur auf Pläne verlässt, vielmehr muss eine Überwachung und Anpassung der Testprozesse stattfinden.
- (sofern relevant) Integrationstests, Ebene B, Teststrategie für Integrationstests
  Im Falle, dass Integrationstests auf High-Level Tests fokussiert sind: ´Dies ist einer der High-Level Tests, der Bestandteil der übergreifenden Teststrategie sein muss.

Optimierungsvorschläge

Fangen Sie damit an, Informationen darüber zu sammeln, was die verschiedenen Tests bewirken. Wichtig hierbei ist, inwieweit Sie einen Einblick über Tiefe und Vollständigkeit der einzelnen Tests erhalten können. Stellen Sie hierbei mögliche Risiken fest.

Häufig beginnt die Verbesserung eines Testprozesses bei einer bestimmten Teststufe und es gibt noch keine Abstimmung mit anderen Teststufen. Legen Sie in einem solchen Fall bei der Strategiebestimmung der jeweiligen Teststufe fest, was die Erwartungen an die anderen Tests sind (Was bewirken diese Tests im Bereich der Testabdeckung?).

Versuchen Sie, offensichtliche „Löcher" oder doppelte Tests aufzudecken und diese zur Diskussion zu stellen.

Führen Sie die Rolle eines Test Koordinators ein, dessen Aufgabe es ist, die verschiedenen Teststufen zu koordinieren, diese in einen Mastertestplan aufzunehmen und die Koordination zu überwachen. Der Testkoordinator berichtet an den Projektmanager und den Auftraggeber des Tests. Um einen Interessenkonflikt zu vermeiden, sollte der Testkoordinator von den zuständigen Gruppen für die verschiedenen Teststufen unabhängig sein.

Prüfen Sie, ob es sinnvoll ist, eine Eingangskontrolle durchzuführen, in der die Testware einer bestimmten Teststufe von einer anderen Partei auf Vollständigkeit und Korrektheit überprüft wird. So wird z. B. die Software eines Steuergeräts (ECU – electronic control unit) zusammen mit dem Testplan, den Testfällen einschließlich der Protokolldateien (falls relevant) und den Testberichten an den Hersteller (OEM - original equipment manufacturer) ausgeliefert. Der OEM evaluiert die Testware um sicherzustellen, dass die Eingangskriterien erfüllt sind.

Ergänzend zu der Prüfung der Testware kann ein so genannter Eingangstest definiert werden und an den Lieferanten des Testobjekts kommuniziert werden. Als Teil des Akzeptanztests für das gelieferte Testobjekt wird dieser Test durchgeführt. Dieser Test zeigt klar und objektiv, ob das Testobjekt reif für die nächste Teststufe ist.

Verwenden Sie Eingangs- und Ausgangskriterien je Teststufe, wobei ausdrücklich nach einer Dokumentation und Nachvollziehbarkeit der Testabdeckung gefragt wird. Verwenden Sie Eingangs- und Ausgangskriterien für jede Teststufe um eine klare Abgrenzung zwischen den unterschiedlichen Teststufen vorzunehmen. Typische Eingangskriterien sind: Es sind keine Fehler des höchsten Schweregrads mehr offen oder das Prototyp-Fahrzeug steht in der definierten Konfiguration zur Verfügung. Typische Ausgangskriterien sind dass alle geplanten Testfälle durchgeführt wurden und keine Fehler mehr offen stehen.

Bei umfangreichen Überlappungen zwischen bestimmten Teststufen gibt es auch die Möglichkeit, beide Teststufen miteinander zu kombinieren. Man denke hierbei an eine Kombination von Systemtest und Abnahmetest zu einem integrierten Test. Kernaspekte dabei sind die Verantwortlichkeiten und Erwartungen. Legitime Gründe für einen integrierten Test sind beispielsweise:

- Die Eignung oder Verfügbarkeit der Testumgebung;
- Die frühzeitige Aufdeckung und insbesondere auch Korrektur von relativ wichtigen Fehlern im Entwicklungsprozess durch den Einsatz von Abnahmetestfällen;
- Der frühzeitige Austausch von Kenntnissen zum System von Seiten der Entwickler und den Auftraggeber;
- Der gemeinsame Einsatz der Testumgebung und der dazugehörigen Verwaltungsverfahren;
- Testtools, die aus Sicherheitsgründen normalerweise nicht in der „produktionsnahen" Umgebung verwendet werden dürfen, stehen - zusammen mit der dabei erforderlichen technischen Unterstützung - dem Auftraggeber zur Verfügung
- Die rechtzeitige Weitergabe von Kenntnissen an den Auftraggeber zum Testen;

- Die frühzeitige Beteiligung beim Testen wird die Auftraggeber zum aktiven Mitdenken über das Testen anspornen sowie die Einführung erleichtern und den Akzeptanzgrad erhöhen;
- Durch den integrierten Test wird der Einsatz von Ressourcen optimiert; Personal und Testeinrichtungen werden zentral eingesetzt und Prioritätskonflikte rechtzeitig verhindert;
- Die Einrichtung des Testens unter einer zentralen Leitung wird das gegenseitige Verständnis erhöhen, wodurch alle Beteiligten besser miteinander kommunizieren können.

Beispiel:

Ein OEM hat den Auftrag zur Entwicklung eines neuen Steuergeräts für ein Bremssystem vergeben. Das ABS des Bremssystems wird ergänzt durch ein elektronisches Bremssystem, das die Handbremse ersetzen kann. Das ABS und das Steuergerät werden von dem Lieferanten entwickelt, der dies zuvor schon für ein Vorgängermodell entwickelt hat. Die Software für das elektronische Bremssystem wird von einem anderen Lieferanten entwickelt. Dieser Lieferant kann den Systemtest nicht auf der Zielhardware durchführen, da diese planmäßig erst zu einem deutlich späteren Zeitpunkt im Projekt bereitgestellt wird. Lieferant und OEM entscheiden, den Systemtest und den Akzeptanztest zusammenzufassen. Der Softwarelieferant wird die Testfälle für den Systemtest erstellen und diese werden vom OEM geprüft. Der Systemtest wird von OEM und Lieferant gemeinsam durchgeführt. Der Lieferant wird diese Tests mit seinen eigenen Testfällen für den Akzeptanztest kombinieren. Durch die Kombination dieser Tests kann die Testumgebung (Fahrzeugprototyp) sehr effizient genutzt werden. Die kombinierte Testdurchführung hat den Vorteil, dass sich das Verständnis des Systems durch den OEM verbessert und dass sich die Fehleranalyse durch die direkte Unterstützung des Lieferanten einfacher gestaltet.
Obwohl der Test mit einem Softwareprodukt starten muss, das nicht die erforderliche Qualität aufweist, um den Akzeptanztest zu starten, hat die Realität gezeigt, dass der kombinierte System- und Akzeptanztest zu einem Produkt geführt hat, das mit einem minimalen Risiko in weniger Zeit durch den OEM akzeptiert wurde, als man benötigt hätte, wären beide Tests separat durchgeführt worden.

1.C Kombinierte Strategie für High-Level Tests und Low-Level Tests oder Prüfungsstufen

Beschreibung

Indem bei der Koordinierung auch die Low-Level Tests oder Prüfungsstufen mit einbezogen werden, bestehen noch mehr Möglichkeiten zur Optimierung.

Die Low-Level Tests haben im Vergleich zu den High-Level Tests folgende Vorteile:
- Sie erfordern wenig Kommunikation, da der Entdecker eines Fehlers häufig auch derjenige ist, der ihn sowohl verursacht hat als auch behebt.
- Sie finden Fehler in einer früheren Phase des Systementwicklungsprozesses als die High-Level Tests.

Verglichen mit Testen bedeutet Prüfen, dass mehr in weniger Zeit und früher im Entwicklungsprozess gefunden wird. Da aber nicht alles geprüft werden kann, bleibt das Testen sehr wichtig. Die Ausdehnung des Abstimmungsbereichs vom Testen auf Prüfen bietet viele zusätzliche Möglichkeiten, um den Aufwand zu optimieren. Insbesondere nicht funktionale Qualitätsmerkmale wie Wartbarkeit, Portabilität oder Verlässlichkeit können vorzugsweise eher geprüft als getestet werden.

Die Einbeziehung von Prüfungen oder Low-Level Tests in die Gesamtteststrategie ist ein Gewinn für den gesamten Testprozess, da dies mehr Möglichkeiten zur Optimierung schafft, d.h., die wichtigsten Fehler werden so früh und so kostengünstig wie möglich gefunden.

Kontrollpunkte

1.C.1 Es findet eine Abstimmung zwischen den High-Level Tests und den Low-Level Tests oder den Prüfungsstufen im Bereich der Teststrategie statt (Risiken, Qualitätsmerkmale, Betrachtungsbereich von Test/Prüfung und Planung).

1.C.2 Das Ergebnis der Abstimmung ist eine koordinierte Strategie, die schriftlich festgehalten wird. Während des gesamten (Prüfungs- und) Testprozesses wird diese Strategie überwacht.

1.C.3 Jeder High-Level Test ermittelt auf der Grundlage der Abstimmung die eigene Teststrategie, wie bei Ebene A beschrieben.

1.C.4 (falls zutreffend) Jeder Low-Level Test ermittelt auf der Grundlage der Abstimmung die eigene Teststrategie, wie beim Kernbereich Low-Level Test, Ebene C, beschrieben.

1.C.5 (falls zutreffend) Jede Prüfungsstufe bestimmt auf der Grundlage der Abstimmung die eigene Prüfungsstrategie, wie beim Kernbereich Prüfen, Ebene C, beschrieben.

1.C.6 Abweichungen im Hinblick auf die koordinierte Strategie werden mitgeteilt, danach erfolgt auf der Grundlage der Risiken eine fundierte Anpassung der Strategie.

Abhängigkeiten

- (falls zutreffend) Low-Level Tests, Ebene C, Strategie für Low-Level Tests
  Low-Level Tests müssen gemäß ihrer eigenen Strategie durchgeführt werden, die durch die koordinierende Teststrategie bestimmt wird.
- (falls zutreffend) Prüfen, Ebene C, Prüfungsstrategie
  Die Prüfungen müssen gemäß ihrer eigenen Strategie durchgeführt werden, die durch die koordinierende Teststrategie bestimmt wird.
- (falls zutreffend) Zeitpunkt der Beteiligung, Ebene C, Aufstellen der Anforderungen
- (falls zutreffend) Integrationstests, Ebene B, Teststrategie für die Integration
  Falls die Integrationstests auf Low-Level Tests ausgerichtet sind: dies ist einer der Low-Level Tests, der dann Teil einer übergreifenden Teststrategie sein kann.

Die Gesamtstrategie muss bereits in einem frühen Stadium bestimmt werden.

Optimierungsvorschläge

Beginnen Sie die Kommunikation über die koordinierte Strategie der High-Level Tests mit der Partei, die für die Low-Level Tests (= Entwickler, meistens auf Seiten des Lieferanten) oder die Prüfungen (= häufig Qualitätssicherung des Lieferanten) zuständig ist. Diese Kommunikation überschreitet fast immer die Grenzen der Organisation. Suchen Sie Überlappungen und „Löcher" in der Deckung zwischen allen Tests und Prüfungen.

Der Auftraggeber muss gemeinsam mit dem Lieferanten des Testobjekts entscheiden, welcher Test wann durchgeführt werden soll. Die Qualität des Systems (oder der Systemkomponenten) lässt sich steigern, wenn früh im Projekt ein Fahrzeugprototyp zur

Verfügung steht, so dass der Lieferant sog. „Rapid Prototyping Techniken" einsetzen kann. Tests, die in dieser Phase des Projekts spezifiziert werden, können später im System- oder Abnahmetest mit der realen Soft- und Hardware genutzt werden. Untersuchen Sie, ob bestimmte High-Level Tests nicht vorzugsweise während der Low-Level Tests oder Prüfungen und umgekehrt ausgeführt werden können. Die Überprüfung auf Normen und Standards der Software (Qualitätsmerkmal: Wartbarkeit) kann beispielsweise Teil eines Tests oder einer Prüfung sein.

Beispiel:

Während der Entwicklung eines adaptiven Geschwindigkeitskontrollsystems (ACC – adaptive cruise control) werden die Tests unter der Verantwortung eines Test Koordinators gemeinsam koordiniert. Das System besteht aus einem ACC-Steuergerät, ESP-Steuergerät, Motorsteuergerät und Getriebesteuergerät, die von vier unterschiedlichen Lieferanten entwickelt werden. Der OEM wird dann die vier Steuergeräte integrieren und das integrierte System testen. Um sicherzustellen, dass er die richtige Qualität erhält, wird gemeinsam mit den Lieferanten eine risikobasierte Teststrategie entwickelt. Hier wird entschieden, worauf sich die Lieferanten bei ihren Tests konzentrieren sollen. Basierend auf dieser risikobasierten Teststrategie werden Eingangs- und Ausgangskriterien definiert. Als Teil der Eingangskriterien wird für jedes Steuergerät ein Eingangstest formuliert.
Dieser Test wird vom OEM durchgeführt und die Beschreibung dieses Tests wird an den Lieferanten kommuniziert. Wenn das Steuergerät den Test nicht erfolgreich durchläuft, wird es an den Lieferanten zurückgesandt. Die Anforderungen werden gemeinsam mit den Lieferanten beschrieben. Die Lieferanten brechen diese Anforderungen in detaillierte Software- und Hardwareanforderungen herunter. Diese Spezifikationen werden auf Basis dessen, was in der risikobasierten Teststrategie beschrieben ist, geprüft. In dieser formellen Überprüfungsrunde ist der OEM direkt involviert. Mit diesem Ansatz minimiert der OEM das Risiko, dass das Steuergerät nicht geeignet ist, um in ein ACC-System integriert zu werden und dass das integrierte System noch immer schwerwiegende Defekte aufweist.

Low-Level Tests und Überprüfungsstufen beginnen früher als High-Level Tests. Bei der Koordination muss man dies berücksichtigen.

Kombinieren Sie die Überprüfung der Testbasis (siehe hierzu auch Kernbereich "Einsatz des Phasenmodells", Ebene B) mit der Überprüfung der Spezifikationen.

Tauschen Sie die Testware zwischen verschiedenen Teststufen untereinander aus, z. B. zwischen Abnahmetest und Modultest. Der Vorteil besteht darin, dass bestimmte Tests nicht doppelt vorbereitet werden müssen. Achten Sie aber darauf, dass dazu geneigt wird, die eigenen Tests nicht vorzubereiten bzw. auszuführen, sodass die Testware die Funktion von Spezifikationen übernimmt. („Wenn die Testfälle fehlerfrei durchgeführt werden, ist das System korrekt realisiert").

1.D Kombinierte Strategie für alle Test- und Prüfungsstufen

Beschreibung

Diese Beschreibung ist die gleiche wie die der vorigen Ebene C. In dieser Ebene findet jetzt aber eine Abstimmung zwischen den High-Level Tests und Low-Level Tests, sowie den Prüfungsstufen statt, um eine weitere Optimierung der gesamten Test- und Prüfungsstrategie zu erreichen.

Kontrollpunkte

1.D.1 Es erfolgt eine Abstimmung zwischen den High-Level Tests, Low-Level Tests und den Prüfungsstufen auf dem Gebiet der Teststrategie (Risiken, Qualitätsmerkmale, Betrachtungsbereich des Tests/der Prüfung und Planung).

1.D.2 Das Ergebnis der Abstimmung ist eine koordinierte Strategie, die schriftlich festgehalten wird. Während des gesamten Prüfungs- und Testprozesses wird diese Strategie überwacht.

1.D.3 Jeder High-Level Test ermittelt auf Grundlage der Abstimmung die eigene Teststrategie, wie bei Ebene A beschrieben.

1.D.4 Jeder Low-Level Test ermittelt auf Grundlage der Abstimmung die eigene Teststrategie, wie bei Kernbereich (Low-Level Tests, Ebene C) beschrieben.

1.D.5 Jede Prüfungsstufe ermittelt auf Grundlage der Abstimmung die eigene Prüfungsstrategie, wie bei Kernbereich „Prüfen", Ebene C, beschrieben.

1.D.6 Abweichungen von der koordinierten Strategie werden gemeldet, wonach abhängig von den Risiken eine fundierte Anpassung der koordinierten Strategie vorgenommen wird.

Abhängigkeiten

- Low-Level Tests, Ebene C, Strategie für Low-Level Tests
  Low-Level Tests müssen gemäß ihrer eigenen Teststrategie ausgeführt werden, die von der koordinierten Strategie bestimmt wird.
- Prüfungen, Ebene C, Prüfungsstrategie
  Prüfungen müssen gemäß ihrer eigenen Teststrategie ausgeführt werden, die von der koordinierten Strategie bestimmt wird.
- Zeitpunkt der Beteiligung, Ebene C, Aufstellen der Anforderungen
  Die Gesamtstrategie muss bereits in einem frühen Stadium bestimmt werden.
- Integrationstest, Ebene B, Teststrategie für die Integration.

Optimierungsvorschläge

Benennen Sie einen Test- und Prüfungskoordinator (TPK), der die Prüfungen und Tests aufeinander abstimmt und diese Abstimmungen überwacht. Der TPK erstattet dem Projektmanager und eventuell auch dem Auftraggeber des Systems Bericht. Wichtig ist die Unabhängigkeit des TPK.

Sorgen Sie dafür, dass der Test- und Prüfungsplan ein integraler Bestandteil des (Systementwicklungs-)Projektplans wird.

Wenn Lieferanten mit Sublieferanten arbeiten, sollte man auch die Teststufen und Prüfungen des Sublieferanten in der übergreifenden Teststrategie berücksichtigen. In diesem Fall können dieselben Prozeduren, die zwischen Auftraggeber und Lieferanten implementiert wurden auch zwischen Lieferant und Sublieferant implementiert werden. Der Lieferant ist dafür verantwortlich, dass der Auftraggeber die richtigen Informationen erhält, um eine übergreifende Teststrategie für den gesamten Entwicklungsprozess seines gewünschten Produkts zu formulieren.

Berücksichtigen Sie die Gesamtkosten und den Nutzen des Testens von der Entwicklung bis hin zum Einsatz im Markt, um eine wohlüberlegte Entscheidung zu fällen, welche Risiken wann gestestet werden sollen, wobei ein optimales Gleichgewicht zwischen Nutzen und Kosten zu erreichen ist.

2 EINSATZ DES PHASENMODELLS

Innerhalb des Testprozesses können mehrere Phasen unterschieden werden, nämlich Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss. Jede Phase setzt sich aus einigen Aktivitäten zusammen. Bei jeder Aktivität ist beschrieben, welches ihr Ziel ist, welches die freizugebenden Produkte sind und wie die Aktivität durchzuführen ist. Die Einteilung in Phasen macht den Testprozess kontrollierbar, da klar wird, wer was wann zu tun hat und wie die verschiedenen Aktivitäten gemeinsam mit den involvierten Parteien geplant und überwacht werden können. Wenn diese Transparenz fehlt, werden Aktivitäten entweder zu spät durchgeführt oder sie werden vergessen. Aktivitäten laufen aus dem zeitlichen Rahmen, da kein Einblick in die Dauer der Aktivitäten zur Verfügung steht. Es besteht keine Möglichkeit zum Einblick in den Fortschritt des Testprozesses und daher ist es kaum möglich, den Prozess innerhalb der vorgesehenen Zeit zu beenden. Schließlich bleibt der Testprozess vermutlich länger als notwendig auf dem kritischen Pfad der Systementwicklung.

2.A Planung, Spezifikation, Durchführung

Beschreibung

Die wichtigsten Phasen im Testprozess sind Planung, Spezifikation und Durchführung.

Die Hauptaktivität der Planungsphase ist die Erstellung eines Testplans. In einem Testplan wird festgelegt, wie, von wem, womit und wann die Testaktivitäten ausgeführt werden sollen.

Die Spezifikationsphase hat das Ziel, die Testfälle aufzustellen und die Testdurchführung vorzubereiten. Ferner wird hier für die Bereitstellung der Infrastruktur gesorgt.

Das Ziel der Durchführungsphase ist die Durchführung der spezifizierten Tests, um Einblick in die Qualität des Testobjekts zu erhalten.

Kontrollpunkte

2.A.1 Für den Test werden (mindestens) folgende Phasen unterschieden: Planung, Spezifikation und Durchführung. Diese werden nacheinander ausgeführt, eventuell je Teilsystem. Hierbei ist eine Überlappung zwischen den Phasen möglich.

2.A.2 Die in jeder Phase auszuführenden Aktivitäten werden nachfolgend genannt. Jede Aktivität ist mit Unteraktivitäten bzw. Aspekten versehen, die als zusätzliche Informationen zu verstehen sind und daher nicht obligatorisch sind.

- Die Planungsphase:

Aktivität	Teilaktivitäten / Aspekte	Produkt
– Formulierung des Auftrags	– Auftraggeber und -nehmer – Betrachtungsbereich – Ziel – Randbedingungen – Ausgangspunkte	– Im Testplan festgelegt

– Festlegen der Testbasis	– Bestimmen der relevanten Dokumentation; wie z. B.: Systemanforderungen, Softwareanforderungen, Softwaredesign und andere Dokumente, die zur Ermittlung von Testfällen herangezogen werden – Identifikation der Dokumentation	– Im Testplan festgelegt
– Festlegen des Betrachtungsbereichs (Was wird getestet und was wird nicht getestet)	–	– Im Testplan festgelegt
– Festlegen der Teststrategie für diese Iteration	– Strategiebestimmung – Erstellen eines Kostenvoranschlags	– Im Testplan festgelegt
– Einrichten der Organisation	– Bestimmung der erforderlichen Funktionen – Zuweisung der Aufgaben, Befugnisse und Verantwortlichkeiten – Beschreibung der Organisation – Zuweisung des Personals – Feststellen der Ausbildungen – Feststellen der Kommunikationsstrukturen – Feststellen der Hierarchie der Berichterstattungen	– Im Testplan festgelegt
– Identifizieren der Testprodukte	– Festlegen der Testprodukte – Erstellen(Auswählen) von Richtlinien	– Im Testplan festgelegt
– Definition von Infrastruktur und Tools	– Definition der Testumgebung – Definition der Testtools – Definition der Büroeinrichtung – Festlegen der	– Im Testplan festgelegt

	Infrastrukturplanung	
– Einrichten des Testmanagements	– Definition Testprozessmanagement (Fortschritt, Qualität, Metriken, Berichte) – Definition Infrastrukturmanagement – Definition Testproduktmanagement – Definition Bearbeitung der Abweichungen	– Im Testplan festgelegt
– Bestimmung der Planung (Aspekte wie Aktivitäten, Abhängigkeiten, Meilensteine, Start und Enddaten sowie benötigte Ressourcen)	– Erstellen einer allgemeinen Planung	– Im Testplan festgelegt
– Festlegung des Testplans	– Feststellen der Risiken, Gefahren und Maßnahmen mit Bezug zum Testprozess – Ermitteln kritischer Testumgebungen (z. B. EMV-Halle, Fahrzeug-Prototyp) – Feststellen des Testplans – Festlegung des Testplans (Genehmigung des Auftraggebers)	– Testplan
– Synchronisieren der Testprozessplanung mit dem gesamten Produktprozess		– Im Testplan festgelegt; Idealerweise ist die Planung in die übergreifende Projektplanung integriert

- Die Spezifikationsphase

Aktivität	**Teilaktivitäten / Aspekte**	**Produkt**
– Aufstellen Testspezifikationen und –skripte	– Testfälle (logisch und konkret) – Definition Ausgangsdateien bzw. Ausgangs-Testdatenbank – Testskripte – Festlegen der Parametereinstellungen	– Testfälle – Definition Dateien bzw. Testdatenbank/ Tabellen für Parametereinstellungen – Testskripte

– Spezifizierung Überprüfung von Testobjekt und -infrastruktur	– Checkliste Testobjekt und Infrastruktur (Vollständigkeits-kontrolle) – Testskript Vorbereitungstest	– Checkliste Infrastruktur – Testskript Vorbereitungstest
– Realisierung Testinfrastruktur	– Testumgebung – Testtools	– Einsatzbereite Testumgebung und Tools

- Die Durchführungsphase:

Aktivität	**Teilaktivitäten / Aspekte**	**Produkt**
– Überprüfung von Testobjekt und –infrastruktur	– Überprüfung Infrastruktur und Testobjekt (Vollständigkeits-kontrolle) – Durchführung des Vorbereitungstests	– Testbares Testobjekt
– Füllen Ausgangsdateien (Startbedingungen für die Ausführung von Testfällen)	– Erfassen der initialen Datensammlung – Einstellen der Parameterwerte	– Ausgangsdateien
– Durchführung der (erneuten) Tests	– Durchführung Testskripte – Durchführung statische Tests (einschl. Beurteilung der Testergebnisse und Analyse der Unterschiede)	– Abweichungen – Testberichte

Abhängigkeiten

- Engagement und Motivation, Ebene A, Zuweisung von Budget und Zeit. Es muss genügend Unterstützung von Seiten des (Projekt-)Managements vorhanden sein, um ein Phasenmodell einzusetzen. Insbesondere die ersten Phasen - Planung und Spezifikation - richten sich nicht auf die Testdurchführung und können daher den Eindruck erwecken, überflüssig zu sein.

Optimierungsvorschläge

Organisieren Sie den Testprozess so, dass er möglichst kurz auf dem kritischen Pfad des Projekts liegt. Meistens gelangt das Testen auf einen kritischen Pfad, wenn der Entwickler die Software für die Testdurchführung geliefert hat. Im Testprozess muss soviel wie möglich bereits im Vorfeld geregelt worden sein, wofür keine zu testende Software erforderlich ist. Damit wird erreicht, dass die Durchlaufzeit des gesamten Projekts so kurz wie nur möglich ist bzw. dass soviel wie möglich Zeit zur Verfügung steht, um alle geplanten Tests durchzuführen. Somit kann die Zeit nach Bereitstellung des Testobjekts (fast) ausschließlich für die reine Durchführung der geplanten Tests genutzt werden.

2.B Vollständiges Phasenmodell Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss

Beschreibung

Beim vollständigen Phasenmodell kommen zwei Phasen hinzu: Vorbereitung und Abschluss.

Eine Hauptaktivität der ist die Detailüberprüfung. Dabei wird die Testbasis auf ihre Testbarkeit hin überprüft. Die Vorbereitungsphase beinhaltet als Hauptaktivität die Detailüberprüfung. Dabei wird die Testbasis auf ihre Testbarkeit hin überprüft. Diese Evaluierung hat folgende Ziele:

- Überprüfung, ob sich die Testbasis für die ausgewählten Testspezifikationstechniken eignet. Wenn nicht, muss man entweder andere Techniken auswählen (was üblicherweise zu einer geringeren Testtiefe und somit weniger Einblick in die Qualität des Testobjekts führt) oder die Testbasis anpassen.
- Einblick in die Qualität der Testbasis in einem möglichst frühen Stadium, so dass bei unzureichender Qualität frühzeitig geeignete Maßnahmen getroffen werden können (Eingangskontrolle).
- Falls die Testbasis vom Auftraggeber geliefert wird, sollte der Lieferant das Ergebnis der Überprüfung der Testbarkeit (testability review) mit dem Auftraggeber diskutieren, um die erforderlichen Maßnahmen einzuleiten.
- Fehler und Mängel in einem möglichst frühen Stadium zu finden, so dass diese nicht erst in der Spezifikationsphase erkannt werden und damit diese Phase verzögern;
- Tester sind mit der Testbasis vertraut zu machen, so dass die Spezifikationsphase schneller und besser verläuft.

In der Abschlussphase stehen zwei Aktivitäten im Mittelpunkt:

- Vervollständigung und Aktualisierung der Testware, so dass diese in anderen Testprozessen wieder verwendbar ist, beispielsweise bei Erweiterungen. Dies reduziert den Aufwand zukünftiger Testprozesse.
- Die Bewertung des getesteten Objekts und des Testprozesses zur Information an den Auftraggeber über die Qualität beider Aspekte und zur Formulierung von Empfehlungen für zukünftige Testprozesse.

Kontrollpunkte

2.B.1 Für High-Level Tests werden ferner folgende Phasen unterschieden: Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss. Die Phasen werden nacheinander ausgeführt, eventuell je Teilsystem. Eine gewisse Überlappung ist möglich.

2.B.2 Im Folgenden werden die auszuführenden Aktivitäten für jede Phase aufgezählt. Jede Aktivität ist mit Unteraktivitäten bzw. Aspekten versehen. Diese dienen als zusätzliche Informationen und sind daher lediglich als Möglichkeiten anzusehen.

- Für die Vorbereitungsphase:

Aktivität	Teilaktivitäten / Aspekte	Produkt
– Detailüberprüfung Testbasis (Überprüfen, ob die Testbasis für die ausgewählten Testspezifikationstechniken geeignet ist)	– Bestimmung der relevanten Dokumentation – Erstellen von Checklisten zur Überprüfung – Beurteilung Dokumentation (Überprüfung) – Erstellen Bericht zur Testbarkeit	– Unstimmigkeiten zur Testbasis – Bericht Testbarkeit

- Für die Abschlussphase:

Aktivität	Teilaktivitäten / Aspekte	Produkt
– Konservieren der Testware – Archivieren der Testware (vollständig und Aktualisieren der Testware in der Form, dass sie für andere (zukünftige) Testprozesse wieder verwendbar ist.)	– Auswahl der zu konservierenden Testware – Sammeln und Aktualisieren der Testware – Übertragung der Testware	– Testware
– Bewertung des Testobjekts	– Bestimmung der offen stehenden Abweichungen und festgestellten Trends – Bestimmung der Risiken bei Freigabe – Formulierung einer Empfehlung	– Freigabeempfehlung, festgelegt im Abschlußbericht
– Bewertung des Testprozesses	– Bewertung der Teststrategie – Planung versus Realisierung	– Festgelegt im Abschlußbericht
– Aufstellen des Abschlußberichts		– Abschlußbericht

Abhängigkeiten

Statische Testtechniken, Ebene A, Überprüfung der Testbasis
Die Überprüfung der Testbasis in der Vorbereitungsphase erfordert den Einsatz einer Technik um aussagekräftig zu sein
Testware Management, Ebene A, internes Testware Management

Damit die Testware nach Beendigung des Tests vervollständigt werden kann, muss sie während des Testprozesses richtig verwaltet worden sein.

Optimierungsvorschläge

Siehe Anweisungen unter „Statische Testtechniken – Ebene A – Detailüberprüfung".
Siehe Anweisungen unter „Testware Management – Ebene C - Übertragbare Testware".

3 ZEITPUNKT DER BETEILIGUNG

Obwohl die tatsächliche Durchführung des Tests normalerweise nach der Realisierung der Software beginnt, kann und muss der Testprozess sehr viel früher anfangen. Eine frühe Beteiligung des Testens bei der Systementwicklung ist dabei behilflich, Fehler so früh bzw. so einfach wie möglich zu finden und sogar zu vermeiden. Zwischen den verschiedenen Tests kann eine bessere Abstimmung stattfinden, und die Zeit, in der das Testen sich auf dem kritischen Pfad im Projekt befindet, kann so minimiert werden.

In diesem Kernbereich werden die Begriffe Testbasis und Anforderungen verwendet. Mit „Testbasis" ist all diejenige Dokumentation gemeint, die zur Ableitung von Testfällen verwendet wird oder verwendet werden kann. Mit „Anforderungen" sind die Kundenanforderungen, Systemanforderungen und / oder das Softwaredesign gemeint. In der modellbasierten Entwicklung wird das Modell oft als Ersatz für das Softwaredesign verwendet. Wird es als Softwaredesign verwendet, stellt das Modell einen Teil der Testbasis dar und wird verwendet, um Testfälle abzuleiten.
Wenn das Modell auf Basis des Softwaredesigns entwickelt, wird und nicht als eigenständiges Model, dann wird das Modell selbst zum Testobjekt.

Das Startniveau ist charakterisiert durch den Start der Aktivität „Testen". Diese Aktivität beginnt unmittelbar vor oder nach dem Moment, wenn die Testdurchführung starten sollte (üblicherweise der Auslieferungszeitpunkt für die Software).

3.A Fertigstellung der Testbasis

Beschreibung

Ein rechtzeitiger Beginn sorgt dafür, dass die Testfälle vorbereitet werden können, bevor das System zum Testen freigegeben wird. Zu diesem Zeitpunkt befindet sich das Testen meistens auf dem kritischen Pfad des Projekts. Da die Tests nur noch ausgeführt zu werden brauchen (sie sind bereits entwickelt worden), wird die Durchlaufzeit des Testens auf dem kritischen Pfad der gesamten Systementwicklung so kurz wie möglich gehalten.

Kontrollpunkte

3.A.1 Die Aktivität „Testen" beginnt entweder zur gleichen Zeit oder früher als die Fertigstellung der Testbasis für einen begrenzten Teil des Systems, das separat getestet werden soll.
(Das System kann in mehrere Komponenten unterteilt worden sein, die getrennt implementiert, fertig gestellt und getestet werden. Das Testen des ersten Teilsystems muss dann gleichzeitig oder früher als die Fertigstellung der Testbasis dieses Teilsystems erfolgen.)

Abhängigkeiten

- Einsatz des Phasenmodells, Ebene A, Planung, Spezifikation, Durchführung
  Ein früherer Start hat nur dann Sinn, wenn die Zeit gut genutzt werden kann. Die Phasen Planung und Spezifikation steuern die Aktivitäten, die durchgeführt werden können, bevor die zu testende Software verfügbar ist.

Optimierungsvorschläge

Machen Sie den Testern und den am Projekt Beteiligten klar, dass die Durchlaufzeit länger ist, wenn die Aktivität „Testen" erst zum Zeitpunkt der Testdurchführung beginnt.

Wenn dann erst noch Testfälle erstellt werden müssen, muss man wählen zwischen einer längeren Durchlaufzeit, einer niedrigeren Qualität der Testfälle oder nicht dokumentierten Testfällen. Engagieren Sie einen Testmanager bei der Festlegung der Testbasis (vorzugsweise eher), der den Testprozess startet.

3.B Aufstellen der Testbasis

Beschreibung

Diese Ebene beinhaltet einen früheren Start (als Ebene A), der eine bessere Abstimmung mit anderen Teststufen darüber ermöglicht, wer was wann testet. Die Abstimmung ist aber weniger sinnvoll, wenn andere Tests bereits in einem fortgeschritteneren Stadium sind. Ein zusätzlicher Vorteil besteht darin, dass die Testvorbereitungen früher beginnen und Fehler dadurch eher gefunden werden.

Kontrollpunkte

3.B.1 Die Aktivität „Testen" startet gleichzeitig oder früher als die Phase, in der die Testbasis (häufig der funktionale Entwurf) aufgestellt wird.

Abhängigkeiten

- Einsatz des Phasenmodells, Ebene B, vollständiges Phasenmodell: Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss
  Die Vorbereitungsphase deckt meistens verschiedene Abweichungen zur Testbasis auf. Es ist wichtig, dass diese Unstimmigkeiten so schnell wie möglich gefunden werden, da die Korrekturkosten dann am niedrigsten sind.

Optimierungsvorschläge

Abstimmung mit anderen Tests (über die Testabdeckung der Qualitätsmerkmale, aber auch über den Betrachtungsbereich des Tests, beispielsweise: Überprüft der Systemtest die Schnittstelle mit dem anderen System oder nicht?)

Stellen Sie sicher, dass eine zeitnahe Detailüberprüfung der fertig gestellten Elemente der Testbasis ausgeführt werden kann.

Engagieren Sie einen Testkoordinator, der die Tests miteinander abstimmt und diese Abstimmung weiterhin überwacht.

Überlegen Sie, anderen Teststufen Einblick in die spezifizierten Testfälle zu geben. Der Vorteil liegt darin, dass andere Tests von diesen Testfällen Gebrauch machen können, um Fehlinterpretationen frühzeitig festzustellen. Abnahmetestfälle können beispielsweise dem Systemtest übergeben werden. Der Systemtest kann dann feststellen (evtl. sogar ohne Software), ob das System gemäß den Testfällen funktionieren wird. Ein mögliches Risiko dabei ist, dass der Systemtest ausschließlich diese Fälle verwendet (was nicht Sinn der Sache ist!) und dass faktisch der gleiche Test zweimal ausgeführt wird.

3.C Aufstellen der Anforderungen

Beschreibung

Wenn das Testen beim Aufstellen der Anforderungen mit einbezogen wird, erhält man mehr Sicherheit über die Qualität des Systems. Das Testen kann sich darauf konzentrieren, dass die Qualitätsanforderungen vollständig und messbar spezifiziert werden, dass Akzeptanzkriterien bestimmt, und die Testbarkeit des Entwurfs und der

Software berücksichtigt werden. Indem solche Klarheit in dieser Phase geschaffen wird, können zu einem späteren Zeitpunkt teure Diskussionen über Anforderungen und Kriterien vermieden werden.

Kontrollpunkte

3.C.1 Die Aktivität „Testen" beginnt gleichzeitig oder früher als die Phase, in der die (Kunden- und System-) Anforderungen gestellt werden.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Beziehen Sie die Tester bei der Aufstellung der Anforderungen mit ein, um dafür zu sorgen, dass die Anforderungen konkret, messbar und testbar sind.

Die wichtigste Voraussetzung ist, dass das Testteam über ausreichende Kenntnisse und Erfahrungen verfügt, um oben genannte Kontrollpunkte ordnungsgemäß zu überprüfen. Sorgen Sie in diesem Zusammenhang für die erforderlichen Sachkenntnisse bzw. Ausbildungen. Die Einrichtung einer Linienabteilung „Testen" vereinfacht den Aufbau und Einsatz von Expertise.

Stellen Sie sicher, dass bei jeder Anforderung Akzeptanzkriterien aufgestellt werden.

Beachten Sie auch die Anweisungen beim Kernbereich „Engagement und Motivation", Ebene C, Testengineering.

3.D Beginn des Projekts

Beschreibung

Die Einbeziehung des Testens in dieser Phase bedeutet, dass bereits vom ersten Augenblick der Systementwicklung an aus der Sicht des Testens über die Auswahl einer bestimmten Entwurfstechnik und -vorgehensweise mitgedacht wird. Wie testbar ist das System bei einer bestimmten Vorgehensweise oder Methode? Wie einfach oder schwierig können nachher Informationen über die Qualität des Systems vermittelt werden?

Kontrollpunkte

3.D.1 Wenn mit der Aufnahme des Projekts begonnen wird, startet auch die Aktivität „Testen".

Abhängigkeiten

- Engagement und Motivation, Ebene C, Testengineering wird akzeptiert
  Die Einbeziehung des Testens ab der Projektaufnahme erfordert ein hohes Maß an Kenntnissen und Erfahrungen vom Testteam. Außerdem muss das Projektmanagement ausreichend viel Vertrauen in die Qualität der Tester haben, um diese bereits in einem so frühen Stadium einzubeziehen.

Optimierungsvorschläge

Beziehen Sie die Tester bereits in den ersten Phasen im Systementwicklungsprozess mit der Aufgabe mit ein, die Testbarkeit des zu entwickelnden Systems zu untersuchen (Faktoren sind hier beispielsweise die auszuwählende Entwicklungsmethode und das Projektkonzept).

Wichtigste Voraussetzung ist, dass das Testteam über ausreichende Kenntnisse und Erfahrungen verfügt, um die oben genannten Kontrollpunkte ordnungsgemäß durchzuführen. Sorgen Sie in diesem Zusammenhang für die erforderlichen Sachkenntnisse bzw. Ausbildungen. Die Einrichtung einer Linienabteilung „Testen" vereinfacht den Aufbau und Einsatz von Expertisen.

Treffen Sie Vereinbarungen mit dem Auftraggeber, welche Testarten von den Lieferanten durchgeführt werden sollen und über die Möglichkeit, aufgrund der Abhängigkeit von Prototypen Fahrzeugen und anderer Testausstattung Tests mit dem Auftraggeber gemeinsam durchzuführen.
Es ist sehr zeitaufwendig, Modelle für HiL (Hardware in the Loop) oder spezielle Simulationsumgebungen zu erstellen. Wenn diese Modelle nicht zur Verfügung stehen, muss der Lieferant so früh wie möglich damit beginnen, die notwendigen Modelle zu erstellen, um seine Testumgebung rechtzeitig zu installieren.

4 KOSTENVORANSCHLAG UND PLANUNG

Die Testplanung und der Kostenvoranschlag geben an, wann welche Aktivitäten auszuführen sind und wie viel Ressourcen (Personal) dafür benötigt werden. Ein qualitativ guter Kostenvoranschlag und eine ebenso gute Planung sind sehr wichtig, da auf dieser Grundlage die erforderliche Kapazität reserviert wird.
Unzuverlässige Kostenvoranschläge und Planungen verursachen häufig eine Verzögerung, da nicht ausreichend viel Ressourcen zugewiesen werden, um die jeweiligen Aktivitäten in der vorgegebenen Zeit auszuführen, oder sie bewirken einen weniger effizienten Einsatz der Ressourcen, weil zuviel davon zugewiesen wurden.

4.A Fundierter Kostenvoranschlag und Planung

Beschreibung

Ein erster wichtiger Schritt bei der Erstellung von Testplanung und Kostenvoranschlag ist die Begründung des Kostenvoranschlags für den Testaufwand. Ein fundierter Kostenvoranschlag verfügt meistens über eine höhere Qualität, da dieser zuverlässiger ist und die Ressourcen effizienter zuweist. Wenn eine Abweichung auftritt, kann besser analysiert werden, ob es sich hierbei um einen einmaligen Fall handelt oder ob die Abweichung einen strukturellen Charakter hat. Im letzten Fall muss wahrscheinlich die gesamte Planung und möglicherweise sogar die Art der Erstellung des Kostenvoranschlags überarbeitet werden. Eine strukturierte Arbeitsweise macht eine kontinuierliche Verbesserung möglich.

Ein optimaler Kostenvoranschlag sowie eine optimale Planung sind sehr wichtig. Auch wenn dies gegeben ist, so ist die Realität doch meist nicht so vorhersehbar, dass die Planung und möglicherweise das Budget ohne jegliche Anpassungen Bestand hätten. Die verschiedensten Maßnahmen müssen ergriffen werden um die Planung und das Budget einzuhalten. Tests oder Testaktivitäten auszulassen ist eine der Maßnahmen, die man vermeiden sollte, da dies Risiken mit sich bringt. Dies führt zu mehr Unsicherheit über die Qualität des Testobjekts.

Im Planungsprozess sollte eine Abstimmung zwischen Auftraggeber und Lieferant stattfinden, um doppelten Aufwand und Lücken im Test zu vermeiden (Siehe auch Kernbereich Teststrategie). Es ist besonders wichtig, die Testunterstützung durch den Lieferanten während Winter- und Sommertests zu planen.

Kontrollpunkte

4.A.1 Der Testkostenvoranschlag und die -planung können begründet werden (aber nicht so:„Wir haben das beim vorigen Projekt genauso gemacht“). Für Basisaktivitäten ist bekannt, welcher zeitliche und finanzielle Aufwand für diese Aktivitäten erforderlich ist.

4.A.2 Im Testprozess erfolgt eine Überwachung des Kostenvoranschlags und der Planung, und falls notwendig findet auch eine Anpassung statt.

4.A.3 Im Fall von kurzfristigen Änderungen, die seitens des Lieferanten und/oder des Auftraggebers erforderlich werden, wird eine Neuplanung der Testaktivitäten durchgeführt.

Abhängigkeiten

- Einsatz des Phasenmodells, Ebene A, Planung, Spezifikation, Durchführung
  Damit ein zuverlässiger Kostenvoranschlag und eine ebensolche Planung aufgestellt werden können, müssen die einzelnen Aktivitäten des Testprozesses unterschieden werden. Hierzu ist ein Phasenmodell erforderlich.

Optimierungsvorschläge

Sammeln Sie Informationen über die (Qualität der) Methode, nach der Kostenvoranschläge und Planungen gemacht werden (verschaffen Sie sich beispielsweise einen Überblick über Kostenvoranschlag und Planung voriger Projekte und ob diese zuverlässig waren).

Versuchen Sie, den Kostenvoranschlag nach unterschiedlichen Gesichtspunkten zu beurteilen. Möglichkeiten zur Bestimmung des Aufwands sind:

- Nehmen Sie einen Prozentsatz des insgesamt veranschlagten Aufwands, der sich auf die Erfahrungen mit ähnlichen Testprozessen gründet (beispielsweise Fachkonzept: 20%, DV-Konzept, Realisierung und Modultest: 40-45%, Systemtest: 15-20%, Abnahmetest: 20%).
- Verwenden Sie Kennzahlen, die auf Erfahrungen mit ähnlichen Testprozessen basieren (unsere eigenen Erfahrungen gehen von folgenden Zahlen aus: 10% Vorbereitung, 40% Spezifikation, 45% Durchführung, einschließlich eines erneuten Tests, 5% Abschluss; die Durchführung eines erneuten Tests kostet etwa 50% der Zeit, die man für eine erste Testdurchführung benötigt, da die Testware jetzt „getestet" und wieder verwendbar ist); veranschlagen Sie den zeitlichen Aufwand für Testmanagementaufgaben auf 10-20%.
- Schätzen Sie die Stunden für die einzelnen Aktivitäten ein, und extrapolieren Sie diese anschließend. Beispielsweise dauert die Spezifikation der Testfälle für eine Funktion 4 Stunden; bei 100 Funktionen sind also 400 Stunden erforderlich. Addieren Sie hierzu geschätzte 50 Stunden für die anderen Aktivitäten in der Spezifikationsphase (Infrastruktur!), und Sie erhalten 450 Stunden. Eine weitere Extrapolation ist jetzt anhand der Standardverhältnisse (siehe voriger Punkt) möglich.
- Extrapolation der Ergebnisse eines Testpilotprojekts
- Umrechnung auf Prozentsätze je Teststufe (Modul-, Integrations-, System- und Abnahmetest)

Erarbeiten Sie ein Verfahren, wie ein Testkostenvoranschlag aufzustellen ist (beispielsweise Anwendung von mindestens zwei Faustregeln).

Verwenden Sie die Ergebnisse anderer Projekte, wo detaillierte Zahlen für die geplanten und die verbrauchten Aufwendungen vorhanden sind.

Beurteilen Sie nach Beendigung des Projekts den Kostenvoranschlag sowie das Verfahren, und passen Sie erforderlichenfalls das Verfahren an.
Treffen Sie bereits vorher entsprechende Vereinbarungen darüber, wie mit dem Anlernen, mit Mehrarbeit und Wartezeiten umzugehen ist.

Berücksichtigen Sie bei der Planung die erforderliche Zeit für:

- Übertragung (von der vorigen Phase, beispielsweise von Systemtest auf Abnahmetest) und Installation des Testobjekts
- Korrektur und erneute Tests
- Denken Sie daran, Zeit, Budget und Ressourcen für die Tests von bisher nicht bekannten Änderungen einzuplanen

In der Praxis hat sich folgende Vorgehensweise für die Planung bewährt: Der gesamte Testprozess wird erst in großen Linien geplant und danach wird jeweils ein Detailplan für die kommenden drei bis vier Wochen aufgestellt.

4.B Statistisch fundierter Kostenvoranschlag und Planung

Beschreibung

Metriken können erstellt und deren Ergebnisse analysiert werden. Auf der Grundlage dieser Analysen kann die Vorgehensweise für Planung und Kostenvoranschlag weiter optimiert werden.

Kontrollpunkte

4.B.1 Es werden Metriken über den Fortschritt und die Qualität (auf Ebene B des Kernbereichs „Metriken") für mehrere vergleichbare Projekte geführt.

4.B.2 Diese Daten werden für die Begründung des Testkostenvoranschlags und der -planung verwendet.

Abhängigkeiten

- Berichterstattung, Ebene B, Fortschritt (Status der Tests und Produkte), Aktivitäten (Zeit und Kosten, Meilensteine), Abweichungen einschließlich Prioritätenzuweisung. Statistisch fundierte Aufwandschätzung und Planung sind nicht sinnvoll, sofern nicht während des gesamten Projekts eine Berichterstattung über den Projektfortschritt stattfindet.
- Metriken, Ebene B, Projektmetriken (Prozess)

Optimierungsvorschläge

Sorgen Sie dafür, dass jedes Projekt anhand von Berichten in Grundzügen den Fortschritt und die Qualität (Fehler) angibt. Später werden hier mehr Details eingebracht, die von der Linienorganisation bestimmt werden. Ein wesentlicher Aspekt hierbei ist das Wachstum der Funktionalität in Bezug auf die anfängliche Planung: Häufig steigt die Funktionalität eines Systems während der Implementierungs- und Testphase noch erheblich, was vielfach an einem ständigen Strom an Änderungsvorschlägen sichtbar ist.

Lassen Sie diese Metriken von der für das Testen zuständigen Abteilung innerhalb der Linienorganisation verwalten und eine periodische Analyse der Metriken ausführen, wobei nach den Kosten/Nutzen-Kennziffern gesucht wird. Welche Systeme hatten viele Probleme in der Produktion, welche weniger? Welche Beziehung kann zu den ausgeführten Tests hergestellt werden, welche mit der Entwicklungsmethode, nach der vorgegangen wurde, usw.? Sorgen Sie dafür, dass auf der Grundlage der oben genannten Informationen Verbesserungsmaßnahmen vorgeschlagen und implementiert werden.

5 TESTSPEZIFIKATIONSTECHNIKEN

Folgende Definition wird verwendet:

Eine Testspezifikationstechnik ist eine standardisierte Methode zum Ableiten von Testfällen aus Ausgangsinformationen.

Durch Anwendung solcher Techniken ist folgendes möglich:
- Der Teststrategie kann eine fundierte Bedeutung gegeben werden, das heißt, die richtige Abdeckung an der richtigen Stelle.
- Fehler können effektiver aufgespürt werden, anstatt „auf gut Glück" Testfälle zu spezifizieren.
- Informationen über die Qualität und Tiefe der Tests entstehen.
- Die Tests sind besser wieder verwendbar.

5.A Nicht formale Techniken

Beschreibung

Die Verwendung von nicht formalen Techniken bedeutet, dass der Person zur Erstellung von Testspezifikationen viel Spielraum für die Aufstellung von Testfällen gelassen wird. Dadurch wird die Qualität des Tests stark von der (Sach-)Kenntnis dieser Person abhängig, und der Deckungsgrad ist in Bezug auf die Testbasis unklar. Diese Vorgehensweise ist jedoch immer noch weit besser, als dass jeder Tester für sich Testfälle ausdenkt, ohne sich um die Dokumentation dieser Tests zu kümmern.

Bei der Spezifikation von Testfällen ist es sehr wichtig, die erwarteten Ergebnisse zu spezifizieren, da die Kontrolle im nachhinein unter dem Zeitdruck der Testdurchführung häufig nicht gründlich genug erfolgt („das Ergebnis ist 990, ich hatte einen Wert zwischen 800 und 1 000 erwartet, also stimmt dies wahrscheinlich").

Kontrollpunkte

5.A.1 Die Testfälle werden anhand der beschriebenen Testspezifikationstechnik aufgestellt.

5.A.2 Die Testspezifikationstechnik erfordert mindestens eine Beschreibung von: a) Anfangssituation b) Verarbeitungsprozess = auszuführende Testaktionen c) erwartetes Endergebnis

Abhängigkeiten

- Keine

Optimierungsvorschläge

Überzeugen Sie die Tester von der Wichtigkeit, die erwarteten Ergebnisse vorab zu überlegen und zu dokumentieren.

Sorgen Sie für eine Beschreibung der Spezifikationstechnik. Versuchen Sie, dabei möglichst viele praktische Anweisungen zu verarbeiten, so dass der Spielraum für den denjenigen, der die Testfälle aufstellt, etwas eingeschränkt wird. Die Testfälle müssen so detailliert beschrieben werden, dass eine andere Person, als die zuständige Person für die

Erstellung von Testfällen, ausreichend viele Informationen bekommt, um die Tests durchführen zu können.

5.B Formale Techniken

Beschreibung

Der Einsatz von formalen Testspezifikationstechniken hat verschiedene Vorteile:

- Eine besser fundierte Entscheidung über die Tiefe und Vollständigkeit des Tests wird möglich;
- Die Testware lässt sich besser wieder verwenden;
- Der Testprozess wird weniger abhängig von der Person, die die Testfälle erstellt und durchführt. Die Testabdeckung des Testobjekts hängt von der gewählten Technik ab:
- Der Testprozess lässt sich besser überwachen, weil im Vorfeld abgeschätzt werden kann, wie viele Testfälle erforderlich sind. Auf diese Weise ist eine bessere Planung und Fortschrittsüberwachung möglich;

Die Einführung mehrerer Techniken ist wichtig, weil verschiedene Systeme (oder Teilsysteme) unterschiedliche Testintensitäten erfordern (Risikobasiertes Testen).

Kontrollpunkte

5.B.1 Neben nicht formalen Techniken werden auch formalere Techniken eingesetzt, wobei der Weg von der Testbasis zu den Testfällen eindeutig vorgegeben ist.

5.B.2 Für High-Level Tests ist eine fundierte Aussage über den Deckungsgrad des Tests ist basierend auf der Sammlung an Testfällen (in Bezug auf die Testbasis) möglich.

5.B.3 Die Testware ist (innerhalb des Testteams) durch die einheitliche Arbeitsweise übertragbar.

Abhängigkeiten

- Testfunktionen und Ausbildungen, Ebene A, Testmanager und Tester
  Die Tester müssen mit den formalen Techniken umzugehen wissen. Das erfordert eine entsprechende Ausbildung und Training.
- Testware Management, Ebene A, internes Testware Management
  Der Einsatz der (relativ teuren) formalen Techniken liefert Testfälle. Äußerst wichtig ist, dass diese Testfälle wieder verwendbar sind, entweder in erneuten Tests oder für den Test späterer Freigaben des Systems. Dies erfordert ein gutes Management der Testfälle.

Optimierungsvorschläge

Sorgen Sie für eine entsprechende Ausbildung und Training der Tester, die mit diesen Techniken arbeiten sollen.

Sorgen Sie für eine Beschreibung der Technik, wenn diese von einer Standardtechnik abweicht.

Eine Vielzahl von Testtechniken ist in der Testliteratur beschrieben [Beizer, 1990], [Binder, 2000], [Kaner u.a., 1993], [Kit, 1995], [Pol u.a., 1999], Notenboom, 2003], d.h., es besteht also keine Notwendigkeit, sie noch einmal neu zu entwickeln.

5.C Mathematische Methoden

Beschreibung

Eine spezielle Kategorie von Testspezifikationstechniken wird aus den Techniken gebildet, die auf Mathematischen Prinzipien basiert. Diese Kategorie lässt sich in zwei Hauptgruppen unterteilen:

- Statistische Anwendungstests
- Formale Methoden

Statistische Anwendungstests stellen eine Form des real-life Testens dar, wo eine Menge von Testfällen abgeleitet wird auf der Basis einer Analyse der (zu erwartenden) Tatsächlichen Systemnutzung. Die Menge der Testfälle muss statistisch repräsentativ sein. [Broekman and Notenboom, 2003].

Formale Methoden sind Methoden, die auf mathematischer Modellierung, Berechnung und Vorhersagen basieren. Sie werden eingesetzt bei der Spezifikation, dem Entwurf, der Analyse, der Erstellung und der Überprüfung von Computersystemen und von Software. Mit formalen Methoden kann eine formale Validierung durchgeführt werden.

Formale Validierung ist der Prozess, Vertrauen zu gewinnen, dass die grundlegenden (Top-Level) Spezifikationen der Anforderungen und Annahmen korrekt sind. Formale Validierung besteht darin, die formalen Spezifikationen herauszufordern indem man Theoreme aufstellt und versucht, diese Theoreme zu beweisen, die sich aus den Anforderungen und Annahmen ableiten lassen.

Formale Verifizierung ist der Prozess, durch formale Deduktion zu zeigen, dass eine Formale Entwurfspezifikation ihren formalen Anforderungsspezifikationen genügt.

Die eingesetzten formalen Methoden hängen von der Notationstechnik ab, die verwendet wird, um die Spezifikationen zu erstellen. Daher ist eine Überprüfung der Testbarkeit der Spezifikationen notwendig, um herauszufinden, ob die gewählte formale Methode auf die zugrunde liegende Spezifikation angewendet werden kann.
Formale Methoden der Ebene 1 sind eine Mischung aus diskreter Mathematik und Logik, Text und Diagrammen. Die Überprüfung dieser Art von Spezifikationen muss die anderen Prüfer überzeugen, dies bedeutet die Prüfung muss sehr sorgfältig dokumentiert werden.
Formale Methoden der Ebene 2 verwenden eine reale Spezifikationssprache, basierend auf den Prinzipien der Logik, diskreter Mathematik und Programmierung
Formale Methoden der Ebene 3 basieren auf Standardlogik und werden von Logikprüfern, Theoremprüfern und Modellprüfern unterstützt

Der Einsatz formaler Methoden hängt von der Notationstechnik ab, die für die Spezifikationen verwendet wird. Daher ist es notwendig, dass es schon früh im Projekt klar ist, dass formale Methoden für die Validierung oder Verifizierung notwendig oder erwünscht sind. Aus diesem Grund muss bei der Erstellung der Spezifikationen eine Notationstechnik verwendet werden, die sich mit formalen Methoden analysieren lässt.

Die Definition und Beschreibung formaler Methoden basiert auf Ruhsby (1995).

Kontrollpunkte

5.C.1 Mindestens eine mathematische Methode wird zur Ableitung von Testfällen eingesetzt

Optimierungsvorschläge

Beginnen Sie mit einer Konsistenzanalyse und Typenprüfung von der Spezifikationen der wichtigsten Systemkomponenten, die mit Ebene 1 – Formalismen erstellt wurden, z. B: Sicherheitsrelevante Komponenten.

Wenn Sie bereits eine HiL Umgebung nutzen, versuchen Sie, basierend auf der Analyse der zu erwartenden Systemnutzung die bestehenden Einzeltestfälle zu Dauertestfällen zu kombinieren und führen Sie auf diese Weise Statistische Anwendungstests ein.

6 STATISCHE TESTTECHNIKEN

Dynamisches Testen ist die Durchführung von Tests anhand einer funktionsfähigen Software. Dies ist jedoch nicht immer erwünscht oder möglich. Wenn man nicht mit ausführbarer Software testet, spricht man von statischem Testen. Bei dieser Form des Testens handelt es sich um die Kontrolle und das Untersuchen von Produkten wie Dokumentation, Verfahren, Quellen usw. Sie zielt mehr auf die Beurteilung von Maßnahmen zum Erreichen einer bestimmten Qualität als auf die Qualität selbst. Die Überprüfung, ob ein Programmierstandard eingesetzt und eingehalten wird, ist ein Beispiel für eine statische Testtechnik. Verschiedene Qualitätsmerkmale können statisch getestet werden (unter anderem Wiederverwendbarkeit, Aktualisierbarkeit, Portabilität, Sicherheit). Statisches Testen ist im Allgemeinen kostengünstiger und früher als dynamisches Testen möglich. Checklisten sind hierfür sehr brauchbare Hilfsmittel.

6.A Detailüberprüfung der Testbasis

Beschreibung

Wie bereits beim Einsatz des vollständigen Phasenmodells beschrieben wurde, ist die Ausführung einer Detailüberprüfung in der Vorbereitungsphase aus drei Gründen wichtig: zur Kontrolle der Testbarkeit, zur frühen Entdeckung von Fehlern in der Testbasis (z. B., wenn eine Funktion falsch spezifiziert ist) sowie zur detaillierten Einarbeitung in die Testbasis. Ein Vorgehen anhand einer Checkliste ist erforderlich, um die Überprüfung auf strukturierte Art auszuführen, so dass man sich auf die wesentlichen Aspekte konzentriert; andernfalls hat man keinen Überblick über jene Aspekte, die berücksichtigt werden müssen und es besteht das Risiko, dass das Dokument nur auf Rechtschreibfehler überprüft wird.

Kontrollpunkte

6.A.1 Vor der Aufstellung der Testfälle wird die Testbarkeit der Testbasis überprüft und dokumentiert.

6.A.2 Bei dieser Überprüfung werden die Checklisten eingesetzt. Die Checklisten beziehen sich auf die Testspezifikationstechniken, die in der Teststrategie ausgewählt wurden.

Abhängigkeiten

- Keine

Optimierungsvorschläge

Machen Sie den Testern die Bedeutung einer Detailüberprüfung der Testbasis und den Einsatz von Checklisten klar.

Legen Sie Checklisten für folgendes an:

- Allgemeine Überprüfungen
  Diese Checkliste beinhaltet allgemein durchzuführende Überprüfungen der Testbasis, beispielsweise „Stimmt das Inhaltsverzeichnis mit dem Rest der Testbasis überein?", „Liegt ein logisches Datenmodell vor?" und „Sind Bildschirm- und Listenlayouts vorhanden?".
- Je verwendete Testspezifikationstechnik:
  Nicht jede Testtechnik eignet sich für eine bestimmte Testbasis. Eine Technik

kann nur verwendet werden, wenn die Testbasis bestimmten Anforderungen entspricht, die für diese Technik spezifisch sind, beispielsweise das Vorhandensein von Zustandsdiagrammen oder eines Datenmodells. Die Festlegung dieser Anforderungen in Checklisten ermöglicht eine schnelle Kontrolle der Eignung der Testbasis für die ausgewählte Testtechnik.

Machen Sie die Ergebnisse der Überprüfung (gefundene Abweichungen, Einblick in die Qualität der Testbasis) für die Tester sichtbar, und weisen Sie darauf hin, dass durch diese Ergebnisse das Verständnis und die Kenntnis der Testbasis erlangt und verbessert werden.

6.B Checklisten

Beschreibung

Checklisten können auch verwendet werden, um nichtfunktionale Qualitätsmerkmale zu testen. Die Checklisten beschreiben die wichtigen Aspekte, auf die man sich während der Überprüfung konzentrieren sollte. Für jeden ist im Voraus klar, was der Schwerpunkt der Prüfung sein soll.

Kontrollpunkte

6.B.1 Statische Tests (über die Überprüfung der Testbasis hinaus) werden anhand von (vom Projektmanagement bzw. Auftraggeber genehmigten) Checklisten durchgeführt. Diese Checklisten werden eingesetzt, um einen statischen Test nichtfunktionaler Qualitätsmerkmale auf dem Testobjekt durchzuführen.

Abhängigkeiten

- Keine

Optimierungsvorschläge

Überzeugen Sie die Tester von der Objektivität und relativen Vollständigkeit der Checklisten als Begründung eines Urteils (im Gegensatz zu Angaben aus der Erinnerung, wie beispielsweise „es ist nicht benutzungsfreundlich").

Überzeugen Sie Tester davon, dass Checklisten bereits im Vorfeld mit den Entwicklern und anderen Projekt beteiligten besprochen werden müssen, um Diskussionen im Nachhinein zu vermeiden oder um eine „stärkere" Position einnehmen zu können.

Sorgen Sie dafür, dass die richtigen (Fach- oder System-)Kenntnisse für die Spezifizierung bzw. Durchführung von statischen Tests vorliegen.

7 METRIKEN

Unter „Metriken" versteht man quantifizierte Beobachtungen der Eigenschaften eines Produkts oder Prozesses, wie zum Beispiel die Anzahl der Zeilen des Quellcodes. Manche Metriken werden aus anderen Metriken zusammengesetzt. Für den Testprozess sind Metriken im Zusammenhang mit dem Fortschritt des Prozesses und der Qualität des getesteten Systems von elementarer Bedeutung. Sie werden dazu eingesetzt, den Testprozess zu überwachen, und sie helfen dabei, Testempfehlungen zu begründen und Systeme oder Prozesse miteinander vergleichen zu können.
Sie helfen des Weiteren bei der Beantwortung von Fragen, weshalb das eine System wesentlich weniger Störungen bei der Produktion aufweist als das andere, oder weshalb der eine Testprozess schneller und gründlicher ist als ein anderer. Bei der Optimierung des Testprozesses sind Metriken besonders nützlich für die Bewertung von Folgen oder Auswirkungen bestimmter Optimierungsmaßnahmen, da sie Informationen von Zeitpunkten vor und nach dem Treffen der jeweiligen Maßnahmen miteinander vergleichen.

Es gibt zwei Vorgehensweisen, anhand derer man Metriken erhalten kann: „top-down" und „bottom-up".

Bei der top-down Vorgehensweise wird von den Wünschen und Anforderungen des höheren Managements ausgegangen. Eine sehr bekannte Form ist das „goal-question metric-conzept" von Basili [Basili u.a., 1984], bei dem auf der Grundlage dieser Anforderungen und Wünsche die Ziele aufgestellt werden. Fragen werden formuliert, um herauszufinden, inwiefern die Ziele erreicht wurden, woraufhin anschließend die Metriken identifiziert werden, um die Fragen zu beantworten. Das ami-Konzept [Pulford u.a., 1995] bietet eine Methode, um einen solchen Einsatz von Metriken zu implementieren.

Nachteil der top-down Vorgehensweise ist, dass das Management nicht immer weiß, welches die richtigen Ziele sind oder es nicht immer mit den Zielen einverstanden ist, z. B., weil zuverlässige Informationen häufig fehlen: ein typisches Huhn-Ei Problem. Auch sind Arbeiter nicht immer besonders enthusiastisch mitzuarbeiten, da das Risiko besteht, dass ihre Leistung in einer für Sie ungünstigen Weise bewertet wird. Schließlich besteht noch das Risiko, dass die Messungen manipuliert werden, um die gesetzten Ziele um jeden Preis zu erreichen.

Bei einer bottom-up Vorgehensweise [Hetzel, 1993] werden die (Zwischen-)Produkte, die Vorgehensweise und die beteiligten Personen als Grundlage für die Messungen betrachtet. Für jedes Zwischenprodukt wird ein Grundstock an Messungen vorgeschlagen:

Input: Informationen über die verwendeten Ressourcen (Menschen, Computer, Tools, andere Produkte usw.) und die Prozessschritte oder Aktivitäten, die ausgeführt werden;

Output: Informationen über die freizugebenden Produkte;

Ergebnis: Informationen über den Einsatz und die Effektivität der freigegebenen Produkte in Bezug auf die gestellten Anforderungen.

Das wichtigste Argument für die bottom-up Vorgehensweise ist, dass die erhaltenen Informationen ausreichen, um fast jede Frage zu beantworten. Diese Methode hilft bei der Feststellung der Ziele für die Testprozessoptimierung.

Bei beiden Vorgehensweisen ist das Engagement des Managements eine notwendige Voraussetzung.

Ohne der top-down Vorgehensweise Abbruch tun zu wollen, haben wir uns bei der Ausarbeitung dieses Kernbereichs für die bottom-up Vorgehensweise entschieden. Diese Methode stützt sich auf bestimmte Basisdaten. Jede Organisation verfügt zur Festlegung dieser Basisdaten über mehr oder weniger ausgeprägte Richtlinien und Verfahren. Hierbei handelt es sich um die elementaren Bausteine, die benötigt werden, um den Einsatz von Metriken zu etablieren. Hier sind die elementaren Bausteine aufgeführt:

- Fortschrittsüberwachung: Veranschlagung und Überwachung von Ressourceneinsatz, Aktivitäten, freizugebenden Produkten, Meilensteinen;
- Konfigurationsmanagement: Überwachung und Verwaltung von Versionen und Änderungen (Quellcode und Entwurfdokumentation);
- Dokumentation der Abweichungen und der Änderungen: Überwachung und Verwaltung von Abweichungen und unbearbeiteten Änderungsvorschlägen (change requests)

Im Folgenden wird für jede Ebene angegeben, worauf sich die Messungen beziehen.

7.A Projektmetriken (über Produkt)

Beschreibung

Für den Testprozess sind Metriken über den Fortschritt des Prozesses und die Qualität des getesteten Systems von großer Bedeutung. Sie werden verwendet, um den Testprozess zu verwalten, die Testempfehlungen zu fundieren, und auch, um Systeme oder Prozesse miteinander vergleichen zu können. Diese Ebene beinhaltet Metriken für den Input (eingesetzte Mittel) und den Output (Ergebnisse usw.).

Kontrollpunkte

7.A.1 Im (Test-)Projekt werden Input Metriken geführt:

- Eingesetzte Ressourcen: Stunden
- Ausgeführte Aktivitäten: Stunden und Durchlaufzeit
- Umfang und Komplexität des zu testenden Systems: Anzahl der Funktionen bzw. Programmieraufwand, Anzahl der Systemanforderungen, Anzahl von Programmzeilen etc.

7.A.2 Im (Test-)Projekt werden Output Metriken geführt:

- Testprodukte: Spezifikationen und Testfälle, Protokolle
- Testfortschritt: ausgeführte Tests,
  Status (erfolgreich beendet /fehlerhaft / nicht beendet)
- Anzahl der Abweichungen: Anzahl der gefundenen Fehler je Teststufe, je Teilsystem, nach Ursache, Priorität, Status (neu, in Lösung befindlich, korrigiert, erneute Test durchgeführt)
- Erreichte Codeabdeckung für zumindest den Low-Level Test, z. B. Anweisungsüberdeckung C0, Zweigüberdeckung C1

7.A.3 Die Metriken werden im Testbericht verwendet.

Abhängigkeiten

- Engagement und Motivation, Ebene B, Testen in Projektorganisation integriert
  Das Führen von Metriken ist als Investition in die Qualität des Testprozesses zu betrachten. Das erfordert bei den Auftraggebern des Testens ein gewisses Maß an Qualitätsdenken.
- Testprozessmanagement, Ebene B, Planung, Durchführung, Überwachung, Anpassung
  Die für Metriken verwendeten Werte müssen zuverlässig sein. Das erfordert eine gute Verwaltung des Testprozesses.

- Berichterstattung, Ebene B, Fortschritt (Status der Tests und der Produkte), Aktivitäten (Zeit und Kosten, Meilensteine), Abweichungen mit Prioritäten
  Das Führen von Metriken hat wenig Sinn, wenn sie nicht festgehalten und verwendet werden.
- Dokumentation der Abweichungen, Ebene A, interne Dokumentation der Abweichungen
  Eine Voraussetzung für den Einsatz vom Metriken ist der Einsatz eines elementaren Abweichungsmanagementsystems, mit dem es möglich ist, Übersichten über die erfassten Abweichungen zu erzeugen.
- (falls zutreffend) Low-Level Tests, Ebene A, Phasenmodell für Low-Level Tests.
  Falls Testabdeckung und andere Code bezogene Metriken verwendet werden sollen

Optimierungsvorschläge

Fangen Sie klein an: Halten Sie die Stunden und Durchlaufzeit der Phasen fest sowie die Anzahl der gefundenen Abweichungen je Phase.

Fangen Sie so früh wie möglich mit dem Messen an, vorzugsweise sogar bevor der Verbesserungsprozess beginnt, so dass später Vergleichsmaterial vorliegt.

Bei einer guten Dokumentation der Abweichungen kann die Anzahl der unterschiedlichen Messungen immer wieder angepasst oder erweitert werden.

Sorgen Sie dafür, dass die Organisation (und nicht jedes Projekt einzeln) an der Festlegung der zu führenden Metriken beteiligt wird.

Implementierung von Metriken wird aufgrund der Auswirkungen auf die Organisation häufig als ein gesondertes Projekt betrachtet. Berücksichtigen Sie dies, und unterschätzen Sie die Problematik nicht. Zu diesem Thema gibt es umfangreiche Literatur.

Verwenden Sie niemals Metriken, um Personen individuell zu überprüfen, zum Beispiel ihre Produktivität. Die Gefahr einer Fehlinterpretation ist zu groß. Ferner fördert dies eine Manipulation der Daten.

Beispiel:

Wir wollen Tester nach ihrer Produktivität bewerten und tun dies auf der Basis der Anzahl abgeschlossener Testfälle je Zeiteinheit. Es stellt sich heraus, dass Tester 1 erheblich weniger Testfälle je Zeiteinheit als Tester 2 erstellt. Der Grund hierfür liegt jedoch darin, dass Tester 1 der bessere Tester als der zweite ist und er daher die komplexesten Tests spezifizieren muss.
Sorgen Sie dafür, dass Metriken als ein permanenter Bestandteil der Dokumentvorlagen für die (Abschluss-)Berichte und Testpläne (zur Begründung von Testkostenvoranschlägen) angesehen werden.

7.B Projektmetriken (über Prozess)

Beschreibung

Neben den Input und Output Metriken der vorigen Ebene werden auch auf dieser Ebene die Ergebnismetriken berücksichtigt: Wie gut testen wir eigentlich? Die Anzahl der gefundenen Fehler alleine ist nicht so aussagekräftig, denn viele gefundene Fehler heißt nicht per Definition, dass gut getestet wurde, denn es kann auch schlecht implementiert worden sein. Andererseits gilt, dass wenige gefundene Fehler bedeuten können, dass gut implementiert, aber unzureichend getestet wurde.

Solche Informationen sind einerseits nützlich, um die Qualitätsempfehlung zu untermauern, und sie können gleichzeitig als Input zur Verbesserung des Testprozesses dienen. Wenn der Testprozess verbessert ist, muss das in irgendeiner Weise kontrollierbar sein. Die Metriken helfen, die Ergebnisse der Verbesserungen sichtbar zu machen.

Um einen klaren Überblick zu bekommen, sollten diese Metriken während der unterschiedlichen Teststufen auf beiden Seiten (Auftraggeber, Lieferant) erfasst werden.

Kontrollpunkte

7.B.1 Im (Test-)Projekt erfolgen Ergebnismessungen für mindestens zwei der folgenden Aspekte:

- Effektivität der Fehlersuche:
  Die gefundenen Fehler im Vergleich zur Gesamtanzahl der vorhandenen Fehler (in %). Letzteres ist schwer messbar, man sollte hier jedoch die gefundene Anzahl an Fehlern in späteren Tests oder in den ersten Monaten nach Produktionsstart berücksichtigen;
  Analyse, welcher Test die Fehler eher hätte finden müssen (dies sagt etwas über die Effektivität voriger Tests aus!);
- Effizienz der Fehlersuche:
  Die Anzahl der gefundenen Fehler pro aufgewandte Stunde, gemessen über die gesamte Testperiode oder über verschiedene Teststufen hinweg;
- Testdeckungsgrad:
  Die Testziele, die mit einem Testfall abgedeckt sind, im Vergleich zur Anzahl der möglichen Testziele (in %). Diese Ziele können sowohl für Funktionsspezifikationen als auch für Softwareanforderungen und das Softwaredesign bestimmt werden, man denke beispielsweise an die Abdeckung von Anweisungen oder von Anforderungen.
- Bewertung der Testware:
  Die Anzahl der gefundenen „Fehler", deren Ursache in einem falschen Testen lagen (z. B. schlechte, nicht fehleraufdeckende Testdaten, falsche Version der Testbasis benutzt), im Vergleich zur Gesamtanzahl der gefundenen Fehler (in Prozent).
- Qualitätsempfinden:
  Ermittelt aus Überprüfungen und Gespräche mit den Anwendern, Testern und anderen Beteiligten, z. B. von Qualitätssicherungsabteilungen zur Verfügung gestellt

7.B.2 Die Metriken werden einschließlich einer Trendanalyse (z. B.: ein vorherbestimmter Kurvenverlauf verglichen mit der tatsächlichen Situation) bei der Testberichterstattung verwendet.

Abhängigkeiten

Berichterstattung, Ebene C, Risiken und Empfehlungen anhand von Metriken belegt
Das Führen von Metriken für den Testprozess hat wenig Sinn, wenn die Metriken weder festgehalten noch verwendet werden. Bei Ebene C der Berichterstattung können solche Metriken eingesetzt werden.

Dokumentation der Abweichungen, Ebene B, umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungsmöglichkeiten. Eine umfangreiche Dokumentation der Abweichungen ist erforderlich, um Metriken für die Qualität des Testprozesses sammeln zu können.

Optimierungsvorschläge

Tools leisten häufig eine gute Unterstützung beim Sammeln von Metriken.

7.C Systemmetriken

Beschreibung

Das Funktionieren eines Systems in der Praxis ist eigentlich die definitive Endkontrolle. Die Erweiterung der Metriken auf den Systemeinsatz anstatt nur für die Entwicklungsphase bietet eine sehr viel höhere Qualität der Informationen. Die Metrikinformation aus der Entwicklungsphase, einschließlich der Protokolle aus Kontrollen und Überprüfungen kann zwar ein sehr positives Bild der Qualität eines Systems vermitteln, wenn jedoch anschließend im produktiven Einsatz eine ganze Reihe von Störungen auftreten, muss dies in das Gesamturteil mit einfließen.

Kontrollpunkte

7.C.1 Die oben genannten Metriken werden für die Neuentwicklung, die Wartung und den Praxiseinsatz geführt.

7.C.2 Die Metriken werden bei der Beurteilung der Effektivität und Effizienz des Testprozesses verwendet.

Abhängigkeiten

- Reichweite der Methodik, Ebene C, Organisationsgenerisch
  Das Führen von Metriken für ein System erfordert, dass die Informationen, die von verschiedenen Seiten angeliefert werden, miteinander vergleichbar sind. Eine generische Testmethode ist hierfür eine Vorbedingung.
- Kommunikation, Ebene C, Kommunikation über die Qualität der Testprozesse auf Organisationsebene
  Das endgültige Ziel bei der Sammlung von Metriken für Systeme besteht darin, zu einer Verbesserung von (Test-)Prozessen zu gelangen. Die Metriken sind daher im Rahmen einer Koordinierungsbesprechung zu erörtern.
- Testprozessmanagement, Ebene C, Überwachung und Anpassung in der Organisation
  Die für die Metriken erforderlichen Daten stammen von verschiedenen Prozessen. Die Zuverlässigkeit dieser Daten ist von größter Bedeutung; das erfordert eine Verwaltung auf Organisationsebene.

Optimierungsvorschläge

Fangen Sie so früh wie möglich damit an, die Effektivität der Fehlersuche (Anzahl der gefundenen Fehler im Test / Anzahl der vorhandenen Fehler nach Produktivstart) und die Effizienz der Fehlersuche (Anzahl der Fehler im Test / Anzahl der Teststunden) festzuhalten. Diese Zahlen können je Projekt bzw. System verglichen werden.

Sorgen Sie dafür, dass die für das Testen zuständige Abteilung innerhalb der Linienorganisation bzw. die Wartungsorganisation die Metriken auf zentraler Ebene verwaltet. Jedes Projekt überträgt die aufgebauten Metriken an diese Linienabteilung.

Die Wartungsorganisation bzw. Linienabteilung Testen beurteilt die Effektivität und Effizienz der Testprozesse.

7.D Organisationsmetriken (>1 System)

Beschreibung

Das eine System verfügt über eine bessere Qualität als das andere. Indem miteinander vergleichbare Metriken eingesetzt werden, können die besseren Systeme erkannt und die Unterschiede analysiert werden. Diese Erkenntnisse können für eine weitere Prozessoptimierung verwendet werden.

Kontrollpunkte

7.D.1 Es werden in der gesamten Organisation miteinander vergleichbare Metriken für die bereits genannten Daten geführt.

7.D.2 Die Metriken werden bei der Beurteilung der Effektivität und Effizienz der einzelnen Testprozesse eingesetzt, um zu einer Optimierung der generischen Testmethode und künftiger Testprozesse zu gelangen.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Die Testabteilung, die für das Testen verantwortlich ist, fordert von verschiedenen Projekten einheitliche Metriken.

Jedes Projekt oder die Wartungsorganisation überträgt die gesammelten Metriken an diese Abteilung.

8 TESTAUTOMATISIERUNG

Ein schnelleres und ein besseres Testen gehört zu den wesentlichen Zielsetzungen vieler Testorganisationen. Die Automatisierung des Testprozesses mit Hilfe von Testtools ist ein wichtiges Hilfsmittel zum Erreichen dieser Zielsetzung. Zur Erläuterung des Kernbereichs „Testautomatisierung" ist es zunächst wichtig festzustellen, was unter einem Testtool verstanden wird:

Ein *Testtool* ist ein automatisiertes Hilfsmittel, das einem oder mehreren Testaktivitäten, wie Planung und Verwaltung, Spezifikation von Testfällen, Aufbau der Ausgangsdateien, Testdurchführung und Beurteilung Unterstützung bietet.

Der *Nachdruck* liegt dabei auf „*unterstützen*". Durch den Einsatz von Testtools muss eine höhere Produktivität bzw. Effizienz erreicht werden können. Das bedeutet, dass ein Testtool erst dann ein Hilfsmittel ist, wenn sein Einsatz einen Vorteil bringt; der Einsatz eines Tools darf kein Ziel an sich sein.

Automatisierung innerhalb des Testprozesses kann auf verschiedene Weisen stattfinden und hat in der Regel eines oder mehrere der folgenden Ziele:
- Weniger erforderliche Stunden;
- Kürzere Durchlaufzeit;
- Höhere Testtiefe;
- Größere Flexibilität beim Testen
- Umfassenderer und schnellerer Einblick in den Status des Testprozesses
- Bessere Motivation des Testpersonals

Bei Planung und Verwaltung können Tools insbesondere bei folgenden Aktivitäten helfen:
- Kostenvoranschlag;
- Planung;
- Fortschrittsüberwachung;
- Konfigurationsmanagement;
- Dokumentation der Abweichungen.

Viele dieser Tools sind nicht spezifisch für das Testen, sondern für das Projektmanagement im Allgemeinen gedacht. Die Tools sind relativ preiswert, einfach zu implementieren, benötigen wenig Einarbeitungszeit und erhöhen die Qualität und Geschwindigkeit der jeweiligen Prozesse.

Bei der Durchführung und Analyse können verschiedene Toolarten Unterstützung bieten:
- Testsequenzer
- Record & Playback
- Load & Stress
- Testabdeckungen
- Codeprüfer, z. B.: um die Konformität mit MISRA zu überprüfen
- Testdatengeneratoren
- Simulatoren
- Drivers und Stubs
- Compiler
- Komparatoren
- Statischer Analysatoren
- Logikanalysatoren
- Debugger
- Monitore

Die Kosten für diese Tools (in Bezug auf Anschaffung, Ausbildungen, Implementierung und Einsatz) sind durchschnittlich höher als die für Planung und Verwaltung, der potentielle Nutzen in Bezug auf Qualität, Geld bzw. Zeit ist jedoch ebenfalls höher. Ein kritischer Erfolgsfaktor für die Automatisierung des Testprozesses mit Hilfe von Testtools ist das Vorhandensein einer strukturierten Testvorgehensweise und Organisation. In einem gut organisierten Prozess können Tools gewiss einen wichtigen Mehrwert darstellen, sie haben jedoch bei einem unzureichend organisierten Testprozess eher kontraproduktive Auswirkungen. Testautomatisierung erfordert die Wiederholbarkeit und Standardisierung der zu unterstützenden Aktivitäten. Ein unstrukturierter Prozess kann diesen Bedingungen nicht entsprechen.

8.A Einsatz von Testtools

Beschreibung

Auf dieser Ebene werden automatisierte Hilfsmittel eingesetzt, die einen offensichtlichen Vorteil bieten.

Kontrollpunkte

8.A.1 Es wurde die Entscheidung getroffen, bestimmte Aktivitäten in den Phasen Planung und/oder Durchführung zu automatisieren. Das Testmanagement sowie die Partei, die die Investition in die Tools tätigt (im Allgemeinen das Linienmanagement oder das Projektmanagement), sind an dieser Entscheidung beteiligt;

8.A.2 Es werden automatisierte Hilfsmittel eingesetzt, die bestimmte Aktivitäten in der Phase Planung und Durchführung unterstützen (z. B. ein Planungstool, ein Tool zur Erfassung der Dokumentation der Abweichungen und/oder selbstprogrammierte Stubs und Drivers, Codeprüfer, MiL, SiL und HiL);

8.A.3 Das Testmanagement und die Partei, die die Investition in die Tools tätigt, erkennen, dass die eingesetzten Tools mehr Vor- als Nachteile bieten.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Setzen Sie vorzugsweise in der Organisation bestehende Tools ein; prüfen Sie, ob diese die Anforderungen erfüllen.

Im Falle von Low-Level Tests stellt die Einführung von Codeprüfern und Abdeckungstools einen ersten Schritt dar, einige Aufgaben innerhalb des Testprozesses zu automatisieren.

8.B Beherrschung der Testautomatisierung

Beschreibung

Auf dieser Ebene wird erkannt, dass die Implementierung, der Einsatz sowie die Verwaltung von Testtools sorgfältig betreut werden muss, da die Wahrscheinlichkeit andernfalls groß ist, dass sich die Investitionen in das Testtool nicht auszahlen. Außerdem wurde festgestellt, ob die automatisierte Testdurchführung realisierbar ist und die gewünschten Vorteile bietet. Bei einer positiven Antwort ist diese Testautomatisierung auch (zum Teil) bereits realisiert.

Beispiel:

Ein Testsequenzer wird bei einem großen Automobillieferanten eingesetzt, um Tests für Steuergeräte mit einer CAN-Schnittstelle zu automatisieren. Schon zu Beginn wird viel Aufmerksamkeit darauf verwendet, diese Architektur wieder zu verwenden und sie so einfach wie möglich zu halten. Der Kern des Sequenzers ist eine Access-Datenbank die durch ein Makro mit dem Inhalt von Word-Dokumenten gefüllt werden kann. Das Format von Testfällen wird durch einen bestimmten Funktionsaufruf mit seinen Parametern und der erwarteten Nachricht, die vom Steuergerät zurückkommt, gebildet. Im ersten Projekt gab es nur die Möglichkeit, jeweils eine Funktion zur Zeit aufzurufen. In der neuen Version ist es möglich, Sequenzen von Funktionsaufrufen zu erstellen und sogar in Echtzeit zu überwachen, was von dem Steuergerät im Test zurückgeliefert wird. Der Sequenzer wird in diesem Projekt mit einem minimalen Investment wieder verwendet und wird auch in einem nächsten Projekt mit seinen neuen Verbesserungen eingesetzt. Die initiale Entscheidung, generell verfügbare Komponenten wie Word und Access einzusetzen war der Grund dafür, dass die Testsequenzer sehr leicht von anderen Projekten als nützliches Instrument akzeptiert wurden. Aufgrund der Dokumentation und der einfach verfügbaren Architektur hat jedes Projekt seine Erweiterungen für diesen Testsequenzer erstellt. Und nicht nur das ursprüngliche Projekt hatte seinen Nutzen von diesem Testsequenzer, auch alle anderen Projekte haben viel Zeit und Geld gespart indem Sie diesen Testsequenzer einsetzten.

Kontrollpunkte

8.B.1 Es wurde eine wohlüberlegte Entscheidung getroffen, welche Komponenten der Testdurchführung zu automatisieren sind. Bei dieser Entscheidung werden jene Arten von Testtools und Testaktivitäten einbezogen, die zur Testdurchführung gehören;

8.B.2 Falls der Beschluss im Zusammenhang mit der Automatisierung der Testdurchführung positiv ausgefallen ist, wird inzwischen bereits ein Tool zur Testdurchführung eingesetzt;

8.B.3 Bei der Einführung des neuen Testtools erfolgt eine Inventarisierung der technischen Aspekte (funktioniert das Testtool auf dieser spezifischen Infrastruktur) und der möglichen Rahmenbedingungen, die an den Testprozess gestellt werden (Testfälle müssen beispielsweise anstatt in einer freien Textform in einer bestimmten Struktur festgelegt werden, so dass ein Testtool diese Struktur als Eingabe verwenden kann);

8.B.4 Falls ein Testsequenzer zur automatisierten Testdurchführung eingesetzt wird, wird bei der Implementierung explizit die Wartbarkeit der aufgenommenen Testskripte berücksichtigt;

8.B.5 Die eingesetzten Tools sind überwiegend in einem zukünftigen Testprozess wieder verwendbar. Dazu ist die Verwaltung der Testtools entsprechend eingerichtet. Mit „überwiegend" ist gemeint, dass Testtools, die ausdrücklich für den Einsatz innerhalb eines einzigen Testprozesses gedacht sind, nicht unbedingt wieder verwendbar zu sein brauchen;

8.B.6 Der Einsatz der Testtools passt in die (gewünschte) Arbeitsweise des Testprozesses, d. h., dass der Einsatz des Testtools keine Ineffizienz oder unerwünschte Einschränkungen des Testprozesses zur Folge hat.

Abhängigkeiten

- Testfunktionen und Ausbildungen, Ebene A, Testmanager und Tester
  Ein guter Einsatz von Testtools für Durchführung und Analyse erfordert Sachkenntnis der Tester.
- Testspezifikationstechniken, Ebene A, nichtformale Techniken
  (Nur dann zutreffend, wenn Testsequenzer verwendet werden.) Eine automatisierte Testdurchführung hat nur Sinn, wenn die Testskripte gut aktualisiert werden können. Das beinhaltet den Einsatz entsprechender Testspezifikationstechniken.

Optimierungsvorschläge

Inventarisieren und untermauern Sie den Bedarf und die Notwendigkeit von Tools. Verwenden Sie dabei nicht nur Produkte, die im Handel erhältlich sind, sondern berücksichtigen Sie, dass auch ganz kleine, selbst zu implementierende Tools wie Stubs, Drivers und Displays im System sehr nützlich sein können. Der Entwickler kann häufig in sehr kurzer Zeit solche Tools implementieren.

Führen Sie einen strukturierten Auswahl- und Einführungsprozess aus.

Sorgen Sie für Ausbildungen und Unterstützung für das Tool, das erworben werden soll.

Führen Sie ein Pilotprojekt durch.

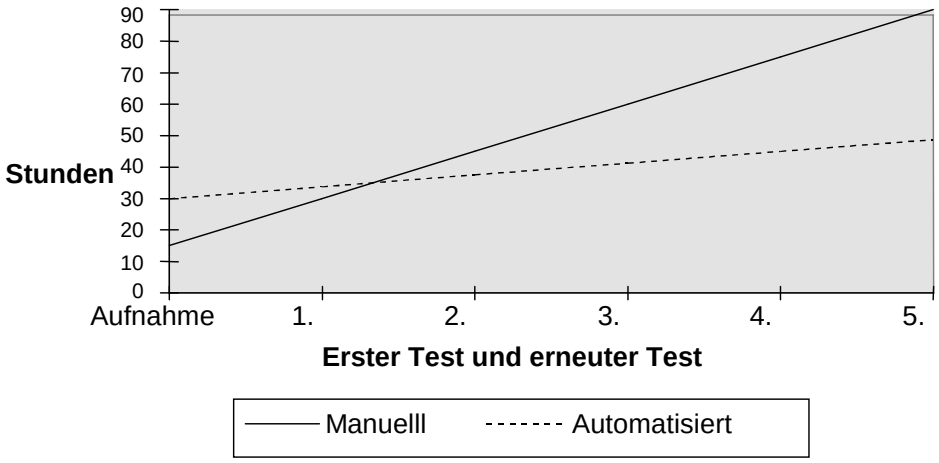
Sorgen Sie dafür, dass sich im Team genügend Sachkenntnis über das Tool befindet (häufig handelt es sich um jemanden, der Technik- und Programmierkenntnisse besitzt).

Erstellen Sie eine Beschreibung, wie die Einrichtung des Tools auszusehen hat.

Erstellen Sie vor der Anschaffung des Tools eine fundierte Kosten/Nutzen-Analyse. Folgendes Beispiel soll einen Eindruck vermitteln, welche Unterschiede zwischen manuellem (M) Testen und automatisiertem Testen mit einem Testsequenzer (A) bestehen:

- Stellen Sie fest, welcher Testaufwand für die Automatisierung in Betracht kommt
  Annahme: Es soll viermal im Jahr ein Regressionstest ausgeführt werden, bei dem vier Personen drei Wochen lang ganztägig mit Testen beschäftigt sind: 4 x 4 x 3 x 5 = 240 Arbeitstage im Jahr
- Schätzen Sie die „reine Ausführungszeit"
  Die „reine Ausführungszeit" ist die Zeit, in der automatisiert werden kann. Es handelt sich um die Zeit, in der jemand vor dem Bildschirm damit beschäftigt ist, Testfälle auf eine Anwendung auszuführen, plus der Zeit, die dafür verwendet wird, Unterschiede festzustellen (das Ergebnis des Tests ist 10, obwohl 9 erwartet wurde). Nicht zur „reinen Ausführungszeit" gehören die Analyse der Unterschiede und die Suche nach der Ursache (die Berechnung ergibt 10 anstatt 9, da in Funktion X ein bestimmter Prozentsatz nicht mitgezählt wird). Im Beispiel von 240 Arbeitstagen jährlich schätzen wir die reine Ausführungszeit auf ein Viertel, also auf 60 Arbeitstage im Jahr.
- Erstellen Sie eine Schätzung für folgende Voraussetzungen
  Die automatisierte Durchführung des ersten Tests kostet durchschnittlich X-mal soviel Zeit wie eine manuelle Durchführung (im Beispiel nehmen wir X = 2, also A = M x 2). Automatisiertes Regressionstesten ist Y-mal schneller als manuelles Testen (im Beispiel nehmen wir Y = 4, A = M/4).
- Berechnen Sie den möglichen Zeitgewinn
  Manuell = 60 Arbeitstage im Jahr oder 15 Arbeitstage je Regressionstest
  Automatisiert = (erster Test kostet doppelt soviel: 15 x 2) + (Regressionstests sind viermal schneller: 3 x 15/4) = 41 Arbeitstage im ersten Jahr und (4 x 15/4) = 15 Arbeitstage in jedem weiteren Jahr
  Nutzen = Unterschied, also 19 Arbeitstage im ersten Jahr und 45 Arbeitstage in

jedem weiteren Jahr
Diese Berechnungen können auch grafisch dargestellt werden, beispielsweise zur Bestimmung des Break Even Points: Nach wie vielen Tests zahlt das Tool sich aus?



- Schätzen Sie die folgenden Faktoren ein: („**-**„ steht für Kosten und „**+**" steht für Nutzen eines Tools)
  - **-** Anschaffung eines Tools;
  - **-** Ausbildungen;
  - **-** Einrichtung des Tools;
  - **-** Aktualisierung der Skripte bei Änderungen (Aktualisierung von automatisierten Skripten ist wesentlich arbeitsintensiver als die Aktualisierung von manuellen Skripten);
  - **+** Höhere Qualität von automatisierten Tests (davon ausgehend, dass der menschliche Tester bei der Durchführung des x-ten Regressionstests weniger aufmerksam wird);
  - **+** Höhere Motivation und Produktivität des Personals (ein Tool verleiht dem Testen eine neue Dimension, es macht mehr Spaß (bei guten Tools))
  - **+** Kürzere Durchlaufzeit

Diese Faktoren müssen eingeschätzt werden, wobei insbesondere die Aktualisierung der automatisierten Skripte viel Aufwand erfordern kann und auch schwierig vorhersehbar ist.

- Erstellen Sie jetzt einen vollständigen Kosten/Nutzen-Vergleich

Der Vergleich steckt zwar voller Annahmen, vermittelt jedoch ebenfalls eine Grundlage für eine weitere Begründung. Häufig stellen sich die Erwartungen als viel zu hoch heraus. Übrigens kann der Vergleich auch für „normale" Tests an Stelle von Regressionstests durchgeführt werden. Berücksichtigen Sie auf jeden Fall, dass die erste Testdurchführung in der Regel zweimal solange dauert wie ein Regressionstest und dass nicht alle Tests einen Regressionstest zur Folge haben werden.

Überwachen Sie die Kosten und Nutzen in regelmäßigen Zeitabständen, wobei angegeben wird, ob der Break Even Point bereits erreicht worden ist oder nicht.

Sorgen Sie dafür, dass Wissen und Erfahrung in Bezug auf das automatisierte Testen gesammelt und verfügbar gemacht werden. Die Einrichtung einer Linienabteilung Testen vereinfacht dies.

8.C Optimale Testautomatisierung

Beschreibung

Man ist sich dessen bewusst, dass Testautomatisierung für alle Testphasen und -aktivitäten eine nützliche Unterstützung bedeuten kann. Dazu wird auf strukturelle Weise untersucht, wo Testautomatisierung einen weiteren Gewinn für den Testprozess bedeuten kann. Das gesamte automatisierte Testen wird periodisch beurteilt.

Kontrollpunkte

8.C.1 Es wird eine wohlüberlegte Entscheidung getroffen, welche Komponenten des Testprozesses zu automatisieren sind. Bei dieser Entscheidung werden alle möglichen Arten von Testtools und Testaktivitäten hinzugezogen;

8.C.2 Es bestehen Einsichten zum Kosten/Nutzen-Verhältnis für alle eingesetzten Testtools (wobei Kosten und Nutzen nicht nur in Geld ausgedrückt zu werden brauchen);

8.C.3 Es erfolgt eine periodische Neubewertung der Vorteile der Testautomatisierung;

8.C.4 Man verfolgt die Entwicklungen auf dem Testtool Markt;

8.C.5 Neue Testtools für den Testprozess werden nach einem strukturierten Prozess eingeführt. Folgende Aspekte innerhalb dieses Prozesses sind zu berücksichtigen:

- Ziele (was soll die Automatisierung in Hinsicht auf Zeit, Geld oder Qualität leisten);
- Bereich (welche Teststufen/Testarten und welche Aktivitäten sind zu automatisieren);
- Erforderliches Personal und Sachkenntnis (eventuell noch erforderliche Ausbildungen);
- Erforderliche technische Infrastruktur;
- Auswahl des Tools;
- Einführung des Tools;
- Entwicklung von wartbaren Skripten;
- Einrichten einer Toolverwaltung.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Organisieren Sie bestimmte strukturelle Aktivitäten, wie die Fühlung mit den Entwicklungen auf dem Testtool Markt in einer unterstützenden Linienabteilung Testen.

Dokumentieren und verwalten Sie den Einführungsprozess und stellen Sie Vorlagen aus der Linienabteilung Testen zur Verfügung.

9 TESTUMGEBUNG

Die Entwicklung eines Steuergeräts für ein Fahrzeug umfasst viele Aktivitäten in unterschiedlichen Stufen im Entwicklungsprozess, wobei jede einzelne wiederum spezifische Testumgebungen erfordert. Typische Komponenten einer Testumgebung sind:

- Vorläufige Hardwareversionen (Prototypen)
- Simulatoren (sowohl für Hardware als auch für Software Komponenten)
- Stubs und Treiber
- Tools, um Inputdaten für den Test zu erzeugen und zu steuern
- Tools, um Outputdaten der Tests zu erfassen und zu kontrollieren
- Einrichtungen für die Ablage und Verwaltung der Testware
- HiL, MiL, SiL
- Breadboards
- Verfahren
- Fahrzeugprototypen

Was genau in einer Testumgebung benötigt wird, hängt von der Stufe im Entwicklungsprozess ab. In frühen Entwicklungsstufen, wenn nur Modelle existieren und noch keine Hard- oder Software, muss die Testumgebung aus einer Modellierungsumgebung mit Möglichkeiten für Statische und Dynamische Tests von Modellen ausgestattet sein. Recht bald darauf, wenn ein Testboard zur Verfügung steht, kann die Software in einer „Rapid Prototype" Umgebung in einem Prototypfahrzeug getestet werden.
In späteren Stufen werden Software und Prototypen der Hardware verfügbar sein und die Testumgebung muss verschiedene Simulatoren enthalten, die andere Systemkomponenten ersetzen und die die Interaktion mit der realen Welt simulieren. Typisch für diese Art von Tests ist die HiL Umgebung.
Letztendlich muss es die Testumgebung erlauben, das „reale System" in seiner „realen Umgebung" zu testen, in dieser Phase wird ein Vorserienfahrzeug als Testfahrzeug verwendet.

Die Qualität der Testumgebung hat großen Einfluss auf Qualität, Durchlaufzeit, und Kosten des Testprozesses. Im Wesentlichen bestimmt sie, ob eine Testaktivität überhaupt durchgeführt werden kann und wie groß der Aufwand dafür ist. Andere wichtige Aspekte der Testumgebung sind: Nutzungsrechte und Zuständigkeiten, Administrierung, Verfügbarkeit, Repräsentativität der Testergebnisse und Flexibilität.

9.A Kontrollierte Testumgebung

Beschreibung

Testen muss in einer kontrollierten Umgebung stattfinden. Unter „kontrolliert" versteht man, dass das Testteam Eigentümer der Umgebung ist (zumindest für die Dauer dieses Testprojekts) und dass die Testumgebung nicht ohne Zustimmung des Testmanagers angepasst wird. Dies verhindert weitgehend, dass Testaktivitäten scheitern und erneut durchgeführt werden müssen, da ein unerwartetes Systemverhalten durch unbekannte und unerwünschte Änderungen in der Testumgebung hervorgerufen wurde. Beispiele solcher Änderungen sind Bugfixes in der Software, Neue Prototypversionen, eine neue Lieferung einer Komponente durch einen (externen) Lieferanten; Ersetzen eines Simulators, Stubs oder Treibers durch eine „reale" Komponente. Wenn das Testteam wiederholt mit solchen Änderungen konfrontiert wird, und keine Kontrolle hierüber hat, wird der Testprozess schnell unkontrollierbar.

Testumgebungen sind typischerweise komplex und brauchen Zeit, erstellt bzw. zusammengestellt zu werden. Um sicherzustellen, dass die Testumgebung rechtzeitig zur Verfügung steht, werden die erforderlichen Vorbereitungsaktivitäten geplant. Dies beinhaltet z. B.: die Entwicklung spezieller Simulatoren, Stubs oder Treiber, die Beschaffung und/oder Konfiguration von Tools, die Koordination von Lieferterminen mit externen Lieferanten für deren Komponenten, das Treffen von Vereinbarungen mit dem OEM über die Verfügbarkeit von Prototypfahrzeugen und deren Minimalspezifikation.

In der Praxis stellt auch das Testobjekt einen Bestandteil der Testumgebung dar und muss daher rechtzeitig bereitgestellt werden. Dies kann bedeuten, dass das Testobjekt eine gewisse Zeit vor Beginn der Testdurchführung zur Verfügung stehen muss, um die Testumgebung fertig zusammenstellen zu können.

Die Winter- und Sommererprobungen können nur während bestimmter Monate im Jahr durchgeführt werden. Den klimatischen Umständen entsprechend gelten spezielle Anforderungen an die Testumgebung. Die richtige Funktionalität des Testobjekts hängt ebenfalls von den klimatischen Gegebenheiten ab.

Die Testumgebung muss so eingerichtet sein, dass die Testdurchführung so effizient wie möglich erfolgen kann. Sofern erforderlich können mehrere Tests parallel ausgeführt werden (falls mehr als eine Testumgebung zur Verfügung steht), ohne die Ergebnisse der anderen Tester zu beeinflussen. Es gibt Mittel und Wege, um die Ausgangssituation eines Tests (Systemzustand und globale Daten) wieder herzustellen, was einen schnellen Start aller Tests (und Nachtests) ermöglicht.

Die Testumgebung sollte so zusammengestellt und repräsentativ sein, wie es die Ziele der jeweiligen Teststufe erfordern. Während früherer Teststufen (typischerweise Integrations- und Systemtests) muss die Testumgebung in Bezug auf die spätere Einsatzumgebung nicht besonders repräsentativ zu sein. Sie sollte jedoch hinreichend Möglichkeiten bieten, detailliert das Systemverhalten zu erfassen und alle Anomalien zu analysieren. In späteren Testphasen – wenn Abweichungen eher die Ausnahme als die Regel sein sollten – fokussiert sich die Zusammenstellung der Testumgebung mehr darauf, der späteren Einsatzumgebung möglichst genau zu entsprechen. Dies bedeutet oft, dass Simulatoren und Tools, die mit dem „echten" Systemverhalten in Konflikt stehen könnten, nicht mehr erlaubt sind.

Kontrollpunkte

9.A.1 Anpassungen bzw. Freigaben an der Testumgebung oder der Testobjekte sind nur mit Genehmigung des Testmanagers erlaubt.

9.A.2 Die Umgebung ist rechtzeitig eingerichtet (Dies kann auch bedeuten, dass das Testobjekt rechtzeitig bereitgestellt werden muss, wenn das Testobjekt einen Teil der Testumgebung darstellt). Im Fall einer eigens entwickelten bzw. erstellten Testumgebung (z. B. Stubkomponenten, etc.) muss ein frühzeitiger Start von Design, Beschaffung, Installation und Konfiguration geplant werden.

9.A.3 Die Testumgebung wird verwaltet (Einrichtung, Verfügbarkeit, Wartung, Versionsverwaltung (Soft- und Hardware), Störfallbehandlung, Autorisierungen, Systemkomponenten, die von Lieferanten und Dritten bereitgestellt werden usw.). Die Konfiguration wird aktualisiert und spiegelt die Erwartungen der nächsten Teststufe wider.

9.A.4 Speicherung und Wiederherstellen bestimmter Testsituationen mit der zugehörigen Version der Testumgebung ist schnell und einfach durchführbar. Im Fall eines Prototypfahrzeugs, das nur für eine begrenzte Zeit zur Verfügung steht,

wird dies später im Projekt kaum möglich sein. In diesem Fall kann dieser Kontrollpunkt unberücksichtigt bleiben.

9.A.5 Die Umgebung ist ausreichend repräsentativ für den auszuführenden Test. Dies hängt vom Testumfang und von der Testart ab. Im Allgemeinen gilt: je weiter man sich im Testverlauf in Richtung Vorserie bewegt, desto mehr muss die Testumgebung der realen Umgebung gleichen.

9.A.6 Die (Hardware und Software) Anforderungen an die Testumgebung sind wohl definiert, verstanden und dokumentiert.

9.A.7 Die Testumgebung, die vom Auftraggeber bereitgestellt wird (z.B.: Prototyp, externe Steuergeräte oder Fahrzeugprototyp) ist gut definiert und dokumentiert.

9.A.8 Auftraggeber und Lieferant sollten die Konfiguration der gemeinsam genutzten und der individuell genutzten Testumgebung koordinieren.

Abhängigkeiten

- Testfunktionen und Ausbildungen, Ebene A, Testmanager, Integrator und Tester
  Eine gute Verwaltung der Testumgebung erfordert geschultes Testpersonal.

Optimierungsvorschläge

Wenn das Bewusstsein für eine funktionsfähige Testumgebung bei den Projektbeteiligten nicht ausreicht, sammeln Sie Beispiele, in denen die Testumgebung „unkontrolliert" war, und diskutieren Sie die dadurch entstandenen Probleme.

Treffen Sie Maßnahmen für einschränkende Faktoren, die nicht geändert werden können. (Beispiel: Wenn von der Lieferung durch das Testteam bis zum Einsatz immer mindestens eine Woche vergeht, beschränken Sie dann die Anzahl der Freigaben, indem Sie andere Aktivitäten, z.B. „creative workarounds" ausführen.)

Sorgen Sie dafür, dass die Verantwortung für die Testumgebung beim Testmanager liegt.

Kümmern Sie darum, dass Backup und Wiederherstellung von Testsituationen, Verwaltung, benötigte Tools usw. rechtzeitig zur Verfügung stehen.

Ein bekanntes Testproblem besteht darin, dass Tests, die in der gleichen Testumgebung ausgeführt werden, einander störend beeinflussen. Um dieses Problem zu umgehen und auch die Durchlaufzeit zu verkürzen, könne Sie überlegen, mehrere Testumgebungen zu organisieren, dann können die Tester simultan arbeiten ohne die Tests der anderen berücksichtigen zu müssen. Ein Nachteil besteht darin, dass die Verwaltung der Testumgebungen komplexer wird und da mehr Testausrüstung benötigt wird, wird das Testen teurer. Jedoch kann hierfür auch ein Schichtbetrieb etabliert werden. (z. B.: führt Team 1 Tests morgens durch, Team 2 führt Tests nachmittags durch)

Sorgen Sie dafür, dass dem Testteam technische Kenntnisse zur Verfügung stehen.

Richten Sie die Testumgebung ein und geben Sie bei Abweichungen die Risiken und eventuellen Maßnahmen an.

Planen Sie gemeinsam mit dem Auftraggeber die rechtzeitige Verfügbarkeit von Fahrzeug Prototypen und welche Anforderungen in diesen Fahrzeugen erfüllt sein sollten
Entwickeln Sie Modelle mit dem richtigen Detaillierungsgrad für HiL Umgebungen. Je detaillierter ein solches Modell ist, desto schwieriger und kostspieliger ist es, ein solches

Modell zu entwickeln. Ein höchst detailliertes Modell ist zum Beispiel früh in der Entwicklungsphase meist nicht nötig. Überlegen Sie für jedes einzelne Modell, welcher Detaillierungsgrad notwendig ist, wie viel Zeit und Budget verfügbar sind und wann das Modell einsatzbereit sein sollte.

9.B Testen in der geeignetsten Umgebung

Beschreibung

Der Aufbau von Testumgebungen und deren Repräsentationsgrad wird heutzutage gut beherrscht. Falls gewünscht, kann das Testteam von der „normalen" Testumgebung abweichen und die Tests in anderen Testumgebungen durchführen, welche besser zu bestimmten Testzielen passen. Auf diese Weise ist es möglich, in einer anderen Umgebung zu testen, die den spezifischen Testzielen besser entspricht oder die eigene Umgebung schnell anzupassen. Dies verschafft mehr Flexibilität bei der Auswahl der optimalen Testumgebung.

Gründe für das Testen in einer anderen Umgebung sind beispielsweise die bessere Eignung dieser Umgebung (z. B. eine schnellere Durchlaufzeit oder die Möglichkeit, das Systemverhalten detailliert analysieren zu können) oder dass ein bestimmter Test früher ausgeführt werden kann. Oft gibt es einen Zielkonflikt: „Detailliertere Manipulation und Analyse des Systemverhaltens" bedeutet üblicherweise „geringere Repräsentativität für die spätere Wirkumgebung". Dabei wird bewusst die Entscheidung getroffen, Testergebnisse frühzeitiger zu erhalten oder eine geringere Repräsentativität zu erzielen. Dies sollte Berücksichtigung finden, wenn man die am besten geeignete Umgebung auswählt.

Im Falle von Fahrzeugprototypen sind die Möglichkeiten begrenzt, etwas anderes als das zu testende Produkt an diesem Fahrzeug zu ändern. Die einzigen Komponenten der Testumgebung, die gewissermaßen kontrollierbar sind, sind die Bedingungen, unter denen der Test durchgeführt wird.

Kontrollpunkte

9.B.1 Jeder Test wird in der Umgebung durchgeführt, die sich am besten dafür eignet: entweder in einer anderen Umgebung (die Umgebung des Lieferanten oder die des Auftraggebers) oder in der eigenen Umgebung die dafür rasch und einfach angepasst wird.

9.B.2 Die Umgebung ist rechtzeitig für den Test vorbereitet, und während des Tests erfolgen keine Störungen durch andere Aktivitäten. Im Fall von Fahrzeug Prototypen sind diese Störungen meist vorhanden, da ein Fahrzeugprototyp weitere Prototyp Steuergeräte enthalten kann, die den Test beeinflussen können. In diesem Fall müssen die Störungen soweit minimiert werden, wie es durch den Tester steuerbar ist.

9.B.3 Die Risiken, die mit angepassten und geänderten Umgebungen eingegangen werden, wurden analysiert, und es wurden angemessene Maßnahmen ergriffen

Abhängigkeiten

- Teststrategie, Ebene B, Kombinierte Strategie für High-Level Tests
  Die Möglichkeit zum Testen in einer anderen Umgebung erfordert eine gute Abstimmung zwischen den verschiedenen Teststufen. Voraussetzung dafür ist also eine koordinierte Teststrategie.

Optimierungsvorschläge

Sorgen Sie dafür, dass das Testen so früh wie möglich beginnt; wägen Sie dabei einerseits die Vorteile einer separaten, verwalteten und repräsentativen Umgebung und andererseits die Vorteile eines frühen Testens bzw. der effizienten Testdurchführung ab.

Beispiel: Teile des Produktionsabnahmetests können in der Systemtestumgebung ausgeführt werden, z.B. Überprüfung des Speicherbedarfs. Ferner können beispielsweise einige wichtige Abnahmetestfälle bereits in der Systemtest oder Entwicklungsumgebung ausgeführt werden, oder es kann die Kontrolle der Benutzungsfreundlichkeit (im Fall von MMI) in der Entwicklungsumgebung stattfinden.

Berücksichtigen Sie die oben genannten Aspekte bei der Bestimmung der Teststrategie, beispielsweise, dass das Testen zwar in einer anderen Umgebung eher beginnt, dass jedoch ein Abschlusstest doch noch in der eigenen Umgebung erfolgen muss.

Stimmen Sie den Einsatz einer (anderen) Umgebung rechtzeitig ab.

Beschränken Sie die Auswahl der Testumgebung nicht auf diejenigen welche innerhalb des Projektteams verfügbar sind sondern berücksichtigen Sie andere Abteilungen und Organisationen des Auftraggebers und des Zulieferers.

9.C Umgebung auf Abruf

Beschreibung

Das Testteam gibt an, welche Tests wann ausgeführt werden müssen und welchen Anforderungen die Testumgebungen genügen muss. Die geforderte Umgebung wird rechtzeitig eingerichtet und freigegeben. Eventuelle Änderungen in der Umgebung sind rasch und flexibel vorzunehmen. Dies bietet dem Testteam eine maximale Flexibilität, um die Testumgebung an geänderte Testziele und sonstige Ereignisse während des Testprozesses anzupassen. In der Praxis lässt sich das nur unter Laborbedingungen realisieren, wie in HiL, SiL, MiL –Umgebungen. Der Fahrzeugprototyp selbst ist nicht sehr flexibel und die Bedingungen lassen sich auch nur sehr begrenzt steuern.

Kontrollpunkte

9.C.1 Die Umgebung, die sich für einen Test am meisten eignet, ist sehr flexibel und kann schnell an geänderte Anforderungen angepasst werden.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Sorgen Sie dafür, dass die verschiedenen Umgebungen gut verwaltet werden.

Diese Ebene stellt in der Tat höhere Anforderungen an die Systembetreuung als an den Testprozess. Berücksichtigen Sie dies!

10 BÜRO- UND LABORAUSSTATTUNG

Das Testpersonal benötigt Räume, Schreibtische, Stühle, PCs, Textverarbeitungsprogramme, Drucker, Telefone, Verbindung mit entfernten Testeinrichtungen usw.

10.A Adäquate und rechtzeitige Einrichtung der Büro- und Laborausstattungen

Beschreibung

Eine gute und rechtzeitige Einrichtung der Büroinfrastruktur hat zur Folge, dass viele mögliche Effizienzverluste wie Umzüge, Wartezeiten und unproduktive Stunden auf ein Minimum beschränkt bleiben. Ferner hat die Einrichtung von Arbeitsplätzen einen positiven Einfluss auf die Qualität des Testprozesses. Man denke in diesem Zusammenhang an die Qualität sowohl der internen als auch der externen Kommunikation sowie an die Motivation und Produktivität der beteiligten Personen.

Eine gute (Daten-)Kommunikationsverbindung zu entfernten Testeinrichtungen ist notwendig, um die Möglichkeit zu schaffen, Testergebnisse von Tests in Fahrzeugprototypen in der Büroumgebung zu analysieren. Von großen Vorteil ist, dass die Büro- und Laborausstattung in der „Zentrale" besser als in verstreuten Laboren dafür geeignet ist, Analysen und Fehlerbehebungen durchzuführen. Die Kommunikationsstruktur vermeidet, dass das gesamte Equipment mehrmals beschafft werden muss.

Kontrollpunkte

10.A.1 Die für das Testen erforderliche Büroinfrastruktur (Büro, Besprechungsräume, Telefone, PCs, Netzwerkanschlüsse, Bürosoftware, Drucker, Kommunikationsverbindungen usw.) ist rechtzeitig bereitzustellen.

10.A.2 Aspekte im Zusammenhang mit der Büroeinrichtung haben eine möglichst geringe Auswirkung auf den Verlauf des Testprozesses (also möglichst wenige Umzüge, keine zu große Distanz zwischen Testern und den restlichen Projektbeteiligten usw.).

10.A.3 Die Büro- und Laborausstattung, die für externe Tests zur Verfügung steht, (z. B: Testwerkzeuge, Fahrzeugtests im Ausland) hat eine durchdefinierte Verbindung zu der Zentrale.

Abhängigkeiten

- Keiner.

Optimierungsvorschläge

Erkundigen Sie sich in einem möglichst frühen Stadium nach den Lieferzeiten der verschiedenen erforderlichen Gegenstände.

Sorgen Sie dafür, dass mögliche Umzüge und dergleichen separat veranschlagt werden.

Wenn Tester in beträchtlicher Entfernung voneinander arbeiten, müssen womöglich zusätzliche Stunden für Gemeinkosten veranschlagt werden. Das verdeutlicht die Nachteile der gewählten Büro- und Laborausstattung.

Berücksichtigen Sie bei der Organisation entfernter Testeinrichtungen die (Daten-) Kommunikationsinfrastruktur mit der „Zentrale”.

11 ENGAGEMENT UND MOTIVATION

Das Engagement und die Motivation der verschiedenen am Testprozess Beteiligten sind wichtige Voraussetzungen für einen gut verlaufenden Testprozess. Dazu gehören nicht nur die Tester selbst, sondern beispielsweise auch das Projektmanagement und das Linienmanagement. Letztere sind insbesondere für die Schaffung guter Voraussetzungen wichtig. Der Testprozess erhält auf diese Weise ausreichende zeitliche, finanzielle und andere (quantitative und qualitative) Mittel, um einen guten Test auszuführen, wobei die Zusammenarbeit und eine gute Kommunikation mit den restlichen Projektmitgliedern einen Gesamtprozess so effizient wie möglich machen.

Ein Merkmal der Startebene ist, dass das Testen als ein notwendiges Übel betrachtet wird. Das Testteam setzt sich zum Großteil aus Personen zusammen, die auf Teilzeitbasis und aus anderen Bereichen (Anwender, Entwickler) zugewiesen werden, um im Testteam mitzuarbeiten. Testen hat einen niedrigen „Status" und wird als überflüssig und ineffizient betrachtet. Die Tester verfügen nur über eine geringe Motivation, vieles wird „pro forma". getestet, und man vertraut voll und ganz auf die Entwickler. Von Testern gefundene Abweichungen werden nicht strukturell bearbeitet.

11.A Zuweisung von Budget und Zeit

Beschreibung

Auf dieser Ebene ist sich das obere und das mittlere Management der Bedeutung des Testens bewusst. Ferner ist man sich darüber im Klaren, dass Testen weder ein geringer noch ein unwichtiger Aufwand ist. Auch sollte das Bewusstsein vorhanden sein, dass die Beteiligung von Repräsentanten des Auftraggebers oder des Lieferanten während der Tests von hohem Wert sein kann bei der Glättung des Prozesses, um mit Leichtigkeit mit Problemen, die während des Testes auftreten, umzugehen.

Wichtig ist, die Mitarbeiter ausreichend zu motivieren, da dies einen direkten Einfluss auf die Produktivität hat. Ein Prozess kann noch so gut eingerichtet sein, ohne gut motivierte Mitarbeiter ist das Endergebnis schlecht. Da Motivation eine kaum messbare Größe ist, sind die Kontrollpunkte in Form verschiedener Aspekte dargestellt, die eine Rolle bei der Motivation spielen.

Kontrollpunkte

11.A.1 Testen wird von den Beteiligten für erforderlich und wichtig gehalten.

11.A.2 Dem Testen wird eine bestimmte Menge an Zeit und Geld zugewiesen.

11.A.3 Das Management lenkt das Testen anhand zeitlicher und finanzieller Mittel. Ein Merkmal ist, dass bei einer Überschreitung der Testzeit oder des Testbudgets hauptsächlich nach einer Lösung innerhalb des Testens gesucht wird (Überstunden oder Einsatz zusätzlicher Mitarbeiter bei einer zeitlichen Überschreitung oder Kürzung von zeitlichen bzw. finanziellen Mitteln der späteren Testaktivitäten).

11.A.4 Im Team sind ausreichende Kenntnisse und Erfahrungen im Bereich des Testens vorhanden, um die Testaufgaben zu erledigen, die dem Testteam zugewiesen wurden. Das Team kann auf Wissen zugreifen, das in Expertengruppen aufgebaut wurde, die sich einem speziellen Thema widmen (z. B. automatisierte Tests, HiL Tests, usw.)

11.A.5 Die Testarbeiten werden von den meisten Teilnehmern in Vollzeit ausgeführt (also bestehen kaum Konflikte mit anderen Aufgaben).

11.A.6 Es gibt klar abgegrenzte Verbindungen zwischen den Testern und anderen Bereichen in Projekt und Organisation.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Dokumentieren Sie die im Einsatz oder in „späten" Tests (Abnahmetest) auftretenden Probleme: Welches sind die Folgen von unzureichenden Testaktivitäten und welches sind die (Organisations-)Kosten, die nach dem Produktionsstart durch die Entdeckung und Beseitigung von Fehlern entstehen?

Überprüfen Sie, welche Fehler man früher hätte finden können.

Halten Sie Vorträge, Seminare usw., um das erforderliche Bewusstsein zu schaffen.

Listen Sie die möglichen Vorteile einer Testabteilung auf und bestimmen Sie, welche Dienstleistungen diese Abteilung anbieten kann.

Sorgen Sie für Schulungen im Bereich des Testens, aber auch auf dem Gebiet von sozialen Fähigkeiten, Systementwicklung, Fachwissen usw.

Arbeiten Sie so schnell wie möglich mit einem Phasenmodell und einer Planung für das Testen. Damit kann der Aufwand optimiert und können Konflikte mit dem Projekt- und dem Linienmanagement vermieden werden, da der erforderliche Einsatz von Mitarbeitern bereits in einem frühen Stadium bekannt ist.

11.B Testen in Projektorganisation integriert

Beschreibung

Ein professionell eingerichteter Testprozess ist besser kontrollierbar und vorhersehbar. Bei einer möglichen Überschreitung (zeitlich oder finanziell) in der Planung besteht mehr Einsicht in die Ursache, so dass entsprechende Maßnahmen getroffen werden können. Ferner besteht eine bessere Kommunikation zwischen den Testern und den übrigen im Projekt beteiligten Gruppen, so dass die einzelnen Gruppen bezüglich der Planung besser aufeinander abgestimmt sind. Dadurch kann der Gesamtprozess effizienter eingerichtet werden.

Da die Produktivität von Mitarbeitern in unmittelbarem Zusammenhang mit ihrer Motivation steht, ist es wichtig, diese so gut wie möglich zu motivieren. Ein gut kontrollierter Testprozess ermöglicht den Mitarbeitern einen guten Einblick auf den aktuellen Projektstatus und zeigt auf, was in der nahen Zukunft von ihnen erwartet wird. Falls der Testprozess schlecht funktioniert sinkt auch die Motivation. Die Anerkennung des Testens durch andere Gruppen sowie Aufstiegsmöglichkeiten für Tester sind weitere motivierende Faktoren.

Beispiel:

Ein hochmotivierter und sehr enthusiastischer Tester, der erfolgreich automatisiertes Testen in drei aufeinander folgenden Projekten implementiert hat, wird zum Testmanager befördert. Diese sogenannte Beförderung bedeutet nichts anderes als dasselbe zu tun, jedoch mit mehr Zusatztätigkeiten. Er beginnt zu realisieren, dass die

Karrieremöglichkeiten als Tester in seiner Firma nicht vorhanden sind. Ihm bleibt nur die Alternative, das Testen zu verlassen und Projektleiter für Entwicklungsprojekte zu werden. Das Traurige daran war, dass es im Bereich Testen vielfältige Karrieremöglichkeiten in seiner Firma gab, aber niemand hatte ihn informiert und die Information wurde von den Mitarbeitern mehr oder weniger geheim gehalten. Indem er ein Projektleiter wurde, hat die Firma einen sehr erfahrenen und in der Vergangenheit äußerst motivierten Tester verloren.

Kontrollpunkte

11.B.1 Alle Beteiligte sind der Ansicht, dass das Testen einen deutlichen und spürbaren Einfluss auf die Qualität des Produkts hat.

11.B.2 Das Management will Einblick in die Tiefe und Qualität des Testens haben.

11.B.3 Das Management lenkt das Testen anhand von zeitlichen, finanziellen und qualitativen Mitteln. Ein Merkmal ist, dass die Lösung für Testprobleme (beispielsweise eine Überschreitung der Testzeit oder des Testbudgets) auch außerhalb des Testprojekts gesucht wird. Dabei wird möglicherweise auch der Entwickler angesprochen.

11.B.4 In der Projektplanung wird der Zyklus Testen, Korrektur und erneut Testen berücksichtigt.

11.B.5 Das Testen beeinflusst die Reihenfolge der vom Entwickler bestimmten Freigabe.

11.B.6 Die Empfehlungen der Tester werden bei der Projektbesprechung erörtert.

Abhängigkeiten

- Einsatz des Phasenmodells, Ebene A, Planung, Spezifikation, Durchführung
- Testprozessmanagement, Ebene B, Planung, Durchführung, Überwachung und Anpassung
  Eine gute Integration des Testens in das Projekt bedeutet, dass der Testprozess planbar und verwaltbar sein muss. Das erfordert den Einsatz eines Phasenmodells und einen gut kontrollierten Prozess.
- Berichterstattung, Ebene B, Fortschritt (Status der Tests und der Produkte), Aktivitäten(Zeit und Kosten, Meilensteine), Abweichungen, einschließlich Prioritäten
- Dokumentation der Abweichungen, Ebene A, interne Dokumentation der Abweichungen
  Eine Mindestvoraussetzung für die Integration des Testens in das Projekt ist, dass der Fortschritt und die Abweichungen gemeldet und richtig verwaltet werden.

Optimierungsvorschläge

Stellen Sie sicher, dass der Testmanager an den Projektbesprechungen teilnimmt.

Sorgen Sie für eine strukturelle Kommunikation und Abstimmung mit dem Entwickler bezüglich des Fortschritts der Teilprojekte.

Schaffen Sie im Testteam eine aktive Einstellung - eine Art Signalfunktion. Die Tester müssen aktiv daran arbeiten, potentielle Qualitätsprobleme so früh wie möglich zu melden.

Beispiel:

Bei einem Projekt herrscht eine sehr straffe Planung. Es wurde vereinbart, dass der Entwickler eine bestimmte Aktivität zu einem bestimmten Datum beendet haben soll, damit am nächsten Tag mit dem Testen begonnen werden kann. Der Entwickler ist immer zum vereinbarten Zeitpunkt fertig, und tags darauf wird die Software auf die Testumgebung übertragen und dort installiert. Diese Übertragung dauert aber - einschließlich der dazugehörigen Probleme - immer zwei bis drei Tage, wodurch das Testen während dieser Tage nicht beginnen kann. Da die Planung oft sehr knapp bemessen ist, wird die Projekt Deadline vom Testteam - als letztes Glied in der Entwicklung - nicht eingehalten. Bei der Projektbesprechung schieben der Entwickler und die Tester in erster Linie einander die Schuld für diese Zeitverzögerung zu, man ist sich jedoch sehr schnell darüber einig, dass die Planung angepasst werden muss. Künftig werden zwei Tage für die Übertragung eingeplant, wonach die Problematik zum größten Teil der Vergangenheit angehört.

Führen Sie im Testteam eine Kommunikation ein, in der das Team über den Fortschritt, die (Test-)Ergebnisse und über das Projekt im Allgemeinen informiert wird.

Fördern Sie die Gründung einer Testabteilung, in der die Kenntnisse und Fähigkeiten der verschiedenen Testteams gesammelt und gebündelt werden können.

Sorgen Sie für Vorträge und dergleichen, um die Organisation bzw. alle Projektbeteiligten von der Bedeutung des Testens zu überzeugen.

11.C Testengineering

Beschreibung

Auf dieser Ebene sind die Motivation, Kenntnisse und Fähigkeiten innerhalb des Testteams so groß, dass der Testprozess als ein wesentlicher Faktor in der gesamten Systementwicklung betrachtet wird, der frühzeitig einbezogen werden muss. Das Testen wird nicht nur als eine Suchmaßnahme, sondern auch als Vorsorgemaßnahme gesehen.

Die zu entwickelnden oder zu wartenden Systeme werden immer komplexer, und der Grad der Integration steigt. Die Testbarkeit der Systeme steht dabei zunehmend unter Druck. Indem man die Testbarkeit beim Entwurf und der Realisierung besser berücksichtigt, kann man diese stark erhöhen, so dass das Testen bei einem geringeren Aufwand mehr Gewissheit über die Qualität des Systems geben kann.

Wenn der Testprozess ein integraler und optimierender Teil des Entwicklungsprozesses ist, dann kann ein Mangel an Qualität so früh wie möglich und zu geringsten Kosten im Gesamtprozess bemerkt oder sogar vermieden werden. Der gesamte Entwicklungsprozess ist dadurch viel besser zu kontrollieren.

Ein optimaler Testprozess erfordert ein hochmotiviertes und ausgebildetes Personal. Neben internen Maßnahmen zum Erreichen einer hohen Motivation bei Mitarbeitern spielen auch verschiedene externe Faktoren eine Rolle. In einem gut eingerichteten Entwicklungsprozess werden Faktoren wie Zeit, Geld und Qualität überwacht. Das verhindert, dass Planungen wiederholt nicht eingehalten werden oder dass die Produkte wie Funktionsspezifikationen und Software eine unzureichende Qualität aufweisen. Wenn das Testteam wiederholt mit solchen Problemen konfrontiert wird, hat das einen negativen Einfluss auf die Motivation. Eine weitgehende Automatisierung hat gerade bei guten Werkzeugen einen positiven Einfluss auf die Tester, weil langweilige und sich wiederholende Arbeiten soweit wie möglich automatisiert sind.

Kontrollpunkte

11.C.1 Beim Entwurf und der Realisierung wird das Testteam mit einbezogen, um eine optimale Testbarkeit des Systems zu erzielen („design for test").

11.C.2 Das Testteam verfügt über ausreichende Kenntnisse und Fähigkeiten, um dem vorgenannten Kontrollpunkt ausreichend Inhalt zu verleihen.

11.C.3 Die Organisation bzw. das Projekt geht mit den Empfehlungen des Testteams „ernsthaft" um.

11.C.4 Das Management unterstützt die Tester (mit Personal und Mitteln), und arbeitet ständig an der Verbesserung des Testprozesses.

11.C.5 Die Teilnahme am Testen wird als „Beförderung" betrachtet; Testen hat einen hohen Status.

11.C.6 Der Entwicklungsprozess ist ausreichend entwickelt; überwacht werden mindestens die Aspekte Zeit und Qualität.

11.C.7 Testfunktionen werden auf Organisationsebene beschrieben, einschließlich der Karrieremöglichkeiten und der Gehaltsordnung.

Abhängigkeiten

- Teststrategie, Ebene C, kombinierte Strategie für High-Level Tests plus Low-Level Tests oder Prüfungsstufen
- Zeitpunkt der Beteiligung, Ebene C, Aufstellen von Anforderungen
- Testtools, Ebene B, Durchführung- und Analysetools
  Testengineering impliziert einen hohen Reifegrad des Testprozesses. Das bedeutet, dass der Testprozess früh in das Projekt mit einbezogen wurde, dass die verschiedenen einzelnen Testarten und Teststufen aufeinander abgestimmt sind und dass der Testprozess ausreichend automatisiert ist.
- Berichterstattung, Ebene C, Risiken und Empfehlungen anhand von Metriken
  Zur Adressierung der in den Kontrollpunkten genannten Empfehlungen ist die Berichterstattung auf einer ausreichend hohen Managementebene erforderlich.

Optimierungsvorschläge

Werben Sie in der Organisation für eine professionelle Testvorgehensweise.

Halten Sie Vorträge für die Entwickler und veranlassen und veranlassen Sie Schulungen im Bereich des Testens.

Sorgen Sie dafür, dass die Funktionen Tester, Testmanager usw. als gesonderte Funktionen bei der Personalabteilung anerkannt werden. Dazu gehört eine umfangreiche Aufgabenbeschreibung.

Veranlassen Sie, dass die Tester (auch) auf ihre spezifischen Testqualitäten hin beurteilt werden.

Bringen Sie beim funktionalen Entwurf und der Realisierung Aspekte ein, die die Testbarkeit erhöhen, beispielsweise eingebaute Beobachtungs- und Kontrollmöglichkeiten.
Beobachtungsmöglichkeiten sind:

- Zeigen von Statusinformationen (interne Programmvariablen);
- Fortschrittsstatistiken;
- Ablaufinformationen (welchen Pfad hat die Software durchlaufen)

Diese Funktionen lassen sich ein- und ausschalten.

Erwägen Sie ebenfalls die Implementierung von in die Software eingebrachten Tests. Diese Tests führen eine bestimmte Standardkontrolle durch, beispielsweise, ob eine Datenstruktur vollständig ausgefüllt ist. Das erhöht das Vertrauen in den Betrieb eines Systems und erleichtert die Fehleranalyse.

Automatisierung des Testprozesses hat einen positiven Einfluss auf die Motivation, da langweilige und sich wiederholende Arbeiten nicht mehr manuell durchgeführt zu werden brauchen.

Für weitgehende Automatisierung siehe die Hinweise bei der entsprechenden Ebene der Testautomatisierung.

Bilden Sie Tester aus, bzw. engagieren Sie Personal mit spezifischen Sachkenntnissen, so dass das Testteam einen sinnvollen Beitrag zum frühen Entwicklungsprozess leisten kann.

Drängen Sie die Organisation dazu, mit einem „Software Process Improvement" Programm zu beginnen, um den Entwicklungsprozess besser kontrollierbar und vorhersehbar zu machen.

Werben Sie in der Organisation für das Testen. Hiermit wird innerhalb der Organisation das Augenmerk auf das Testen gerichtet, auf die Vorteile des Testens, darauf, dass es ein separates Fach ist, usw.

12 TESTFUNKTIONEN UND AUSBILDUNGEN

In einem Testprozess ist die richtige Zusammenstellung eines Testteams sehr wichtig. Es bedarf einer Mischung von verschiedenen Bereichen, Funktionen, Kenntnissen und Fähigkeiten. Neben der spezifischen Testsachkenntnis sind beispielsweise auch die Kenntnisse der zu testenden Materie (Fachwissen) sowie Kenntnisse der Organisation und zur Softwareentwicklung notwendig. Ferner sind soziale Fähigkeiten sehr wichtig. Eine solche Mischung erfordert unter anderem eine entsprechende Ausbildungen.

12.A Testmanager, Integrator und Tester

Beschreibung

Der Einsatz von fachkundigen Testern ist für einen gut verlaufenden Testprozess von großer Bedeutung. Bei einem Testteam, das sich beispielsweise ganz aus Anwendern oder Entwicklern zusammensetzt, ist die Wahrscheinlichkeit viel kleiner, dass ein qualitativ guter Testprozess zustande kommt.

Abgesehen von den spezifischen Kenntnisse und Fähigkeiten, über die ein Tester verfügen muss, spielt hier auch ein bisschen „Psyche" mit. Myers [Myers, 1979] hat bereits nachgewiesen, dass die Grundeinstellung eines Testers eine wesentlich andere ist als die eines Entwicklers: Ein Tester versucht, den Mangel an Qualität aufzuzeigen, und begibt sich dazu aktiv auf die Suche nach Fehlern. Ein Entwickler hingegen will vielmehr nachweisen, dass das System gut ist.

Kontrollpunkte

12.A.1 Das Testpersonal besteht aus mindestens einem Testmanager und einigen Testern. Wenn es ein Bestandteil des Testprojekts ist, Lieferungen von Zweiten oder von Dritten zu integrieren, muss zusätzlich die Rolle eines Integrators definiert werden.

12.A.2 Die Aufgaben und Zuständigkeiten einschließlich der erforderlichen Expertise und möglicher Schulungen sind festgelegt.

12.A.3 Das Testpersonal hat eine spezifische Testausbildung absolviert (beispielsweise Testmanagement und Testspezifikationstechniken) oder verfügt über ausreichende Erfahrungen auf dem Testgebiet.

12.A.4 Für den Test stehen dem Testteam die Fachkenntnisse aus dem Anwendungsbereich zur Verfügung. Dies gilt insbesondere für den Abnahmetest.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Sorgen Sie dafür, dass Tester, Testintegrator und Testmanager entsprechende Testausbildungen absolvieren.

Nehmen Sie im Testplan Funktionsbeschreibungen auf, in denen angegeben ist, wer welche Aufgaben ausübt.

Kümmern Sie sich rechtzeitig um die erforderlichen Fachkenntnisse (insbesondere bei Abnahmetests).

12.B (Formale) methodische, technische und funktionale Unterstützung, Management von Testprozess, Testware und Infrastruktur

Beschreibung

Neben dem Einsatz von sachkundigen Testern ist es wichtig, im Testprozess Unterstützung und Management zu organisieren. Bei einer methodischen Unterstützung muss beispielsweise an die Einrichtung des Testprozesses gedacht werden, wie etwa das Aufstellen von Vorschriften und die Bestimmung der Teststrategie. Technische Unterstützung ist erforderlich, um die Infrastruktur einzurichten und zu bedienen. Funktionale Unterstützung hilft bei der Beantwortung von Fragen zur Funktion, die während des Testens entstehen. Insbesondere für die bei den letzten Formen der Unterstützung sind Personengruppen von außerhalb des Testprozesses erforderlich, wodurch es notwendig ist, den Einsatz gut und rechtzeitig zu regeln. Eine unzureichende Unterstützung kann den Testprozess erheblich verzögern.

Außerdem wird der Kontrolle der eigenen Testprodukte viel Aufmerksamkeit geschenkt: Quality Assurance (Qualitätssicherung). Das soll verhindern, dass Abweichungen bzw. Entgleisungen des Prozesses zu spät entdeckt werden. Werden diese Probleme nicht rechtzeitig beseitigt, hat das fast immer seinen Preis: entweder zusätzliche Zeit oder geringere Testqualität. Im Falle von kleinen Teams kann die hier genannte Form der Unterstützung von großer Hilfe sein. Dennoch muss in kleinen Teams die Unterstützung aufgrund von Kapazitätsproblemen auf einem minimalen Niveau gehalten werden. Unterstützung wird also nur dann in Anspruch genommen, wenn sie zwingend erforderlich ist und auch dann nur für eine klar begrenzte Zeit.

Beispiel:

Unerfahrene Tester können viele Fehler bei der Testspezifikation machen. Wenn das rechtzeitig entdeckt wird, kann man problemlos korrigierend eingreifen, beispielsweise durch zusätzliche Trainings und begrenzte Überarbeitungen. In einem späteren Stadium ist der Schaden sehr viel schwieriger zu beheben.

Kontrollpunkte

12.B.1 Die Aufgabe der methodischen Unterstützung ist gesondert festzulegen. Zu den Aktivitäten gehören das Aufstellen und Aktualisieren von Testvorschriften, -verfahren und –techniken, sowie die Beratung und Kontrolle über der korrekten Anwendung.

12.B.2 Die Aufgabe der technischen Unterstützung ist gesondert festzulegen.

12.B.3 Die Aufgabe der funktionalen Unterstützung ist gesondert festzulegen.

12.B.4 Die Aufgaben der Testprozessverwaltung sind gesondert festzulegen. Zur Testprozessverwaltung zählt das Erfassen, Speichern und Verfügung stellen aller Verwaltungsobjekte des Testprozesses. Dies erfolgt manchmal durch eigenständiges Führen der Verwaltung, in anderen Fällen durch Einrichtung bzw. Kontrolle der Verwaltung. Zu verwaltende Objekte sind Fortschritt, Budgets und Abweichungen.

12.B.5 Die Aufgaben des Testware Managements sind gesondert festzulegen. Zu den Aufgaben des Testware Managements zählt die Erfassung, Speicherung und zur

Verfügung stellen aller Verwaltungsobjekte der Testware. Dies geschieht manchmal durch eigenständiges Führen der Verwaltung, in anderen Fällen durch Einrichtung bzw. Kontrolle der Verwaltung. Zu verwaltende Objekte sind Testdokumentation, Testbasis und Testobjekt (intern), Testware einschließlich Dateien, Testanweisungen und Prozeduren.

12.B.6 Die Aufgabe Management der Testinfrastruktur ist gesondert festzulegen. Sie besteht aus der Erfassung, Speicherung und zur Verfügung stellen aller Verwaltungsobjekte der Testinfrastruktur. Dies erfolgt manchmal durch eigenständiges Führen der Verwaltung, in anderen Fällen durch Einrichtung bzw. Kontrolle der Verwaltung. Zu verwaltende Objekte sind Testumgebungen und Testtools.

12.B.7 Die Personen, die die oben genannten Aufgaben erfüllen, verfügen über ausreichende Kenntnisse und Erfahrungen.

12.B.8 Die für die genannten Aufgaben erforderliche Zeit wird eingeplant. Es wird kontrolliert, ob die Aufgaben tatsächlich durchgeführt werden.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Nehmen Sie im Testplan Funktionsbeschreibungen auf, in denen angegeben ist, wer welche unterstützenden Aufgaben hat.

Reservieren Sie für die unterstützenden Aufgaben entsprechende Kapazitäten, und überprüfen Sie, ob diese Kapazitäten tatsächlich genutzt werden.

Beginnen Sie mit der Kontrolle der korrekten Einhaltung der Vorschriften, Verfahren und Techniken, indem Sie Testern oder Testmanagern einfache Kontrollaktivitäten zuweisen, beispielsweise die Überprüfung der gegenseitigen Testspezifikationen usw.

Nutzen Sie die Unterstützung im Falle kleiner Teams für spezialisierte Aufgaben und nur für begrenzte Zeit. Die Unterstützung wird immer Kapazitäten der Teammitglieder binden. Eine permanente Mitarbeit im Team kann in kleinen Teams nicht für alle unterstützenden Rollen realisiert werden.

12.C Formale interne Prüfung

Beschreibung

Die bei der vorigen Ebene beschriebene Kontrolle der Testaktivitäten wird erweitert und formalisiert, um zwei Ziele zu erreichen:

- Vertrauen in die Qualität des Testprozesses schaffen; das ist gleichbedeutend mit der Einsicht, dass Testen notwendig ist: dies schafft ein größeres Vertrauen in die Qualität des getesteten Objekts.
- Die Ermöglichung einer ständigen Verbesserung des Testprozesses

Kontrollpunkte

12.C.1 Parallel zum Testplan wird ein interner Review Plan für das Testen aufgestellt.

12.C.2 Die mit dem Review der Tests beauftragte Person erfüllt keine weiteren Aufgaben innerhalb des Testteams.

12.C.3 Die Ergebnisse aus dem Review der Testaktivitäten fliesen in die Optimierung des Testprozesses mit ein.

12.C.4 Die Person, welche die Reviews durchführt, verfügt über ausreichende Kenntnisse und Erfahrungen im Bereich des Prüfens.

Abhängigkeiten

- Reichweite der Methodik, Ebene A, projektspezifisch
  Das Prüfen beinhaltet, dass der Testprozess beurteilt und überprüft wird. Diese Aufgabe ist nur sinnvoll, wenn davon ausgegangen werden kann, dass der Testprozess gemäß eines beschriebenen Prozesses durchgeführt wird.

Optimierungsvorschläge

Sorgen Sie dafür, dass die Rolle „internen Prüfung" von einem Mitarbeiter der Testabteilung ausgeführt wird.

Achten Sie darauf, dass die Ergebnisse der Prüfungen tatsächlich entweder vom Testmanager oder von der Testabteilung verwendet werden (beispielsweise für eine weitere Testprozessoptimierung).

13 REICHWEITE DER METHODIK

Jeder Testprozess in der Organisation findet gemäß einer bestimmte Methode oder Vorgehensweise mit bestimmten Aktivitäten, Verfahren, Vorschriften, Techniken usw. statt. Wenn diese Methoden stets projektabhängig oder noch schlimmer von Personen abhängig sind, verliert man wertvolle Zeit, um einen Testprozess für ein neues Projekt zu etablieren. Wenn die Methode ausreichend generisch ist, und auch von anderen als nur den Projektmitgliedern verstanden wird, könne Testprozesse schnell organisiert werden. Eine gut beschriebene Methodik erhöht die Möglichkeit des Lernens und der Verbesserung. Die Methodik muss hinreichend viele Informationen liefern, um jeden Testprozess in der Organisation so zu organisieren, dass aus projektspezifischen Gründen nur kleine Ergänzungen zu der Methodik erforderlich sind.

Die Natur der Automobilindustrie macht es erforderlich, dass die Methodik nicht nur Arbeitsweisen innerhalb des Testteams beschreibt. Ein sehr wichtiger Teil ist der Umgang mit Lieferanten und/oder dem Auftraggeber. Dies gilt sowohl intern als auch extern.

13.A Projektspezifisch

Beschreibung

Der Testprozess, eine Auswahl von Aktivitäten, Arbeitsabläufen, Anweisungen etc. muss gut dokumentiert sein. Die Qualität des Testprozesses wird dadurch weniger von individuellen Personen abhängig, die Anlernzeit für neue Mitarbeiter wird verkürzt und die Eindeutigkeit der Methode erhöht die Qualität des Prozesses. Nur eine dokumentierte und strukturierte Arbeitsweise bietet die Möglichkeit einer kontinuierlichen Verbesserung.

Kontrollpunkte

13.A.1 Die Methodik wird für jedes Projekt formuliert.

13.A.2 Die beschriebenen Aspekte beinhalten zumindest folgendes: Beschreibung des vollständigen Phasenmodells des Testens, Verwaltung des Testprozesses (Fortschritt und Qualität), Testproduktmanagement, Verwaltung von Abweichungen, und anzuwendende Testspezifikationstechniken.

13.A.3 Es wird tatsächlich nach der Methodik vorgegangen.

Abhängigkeiten

- Einsatz des Phasenmodells, Ebene B, Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss
- Testspezifikationstechniken, Ebene B, formale Techniken
- Testware Management, Ebene A, internes Testware Management
- Dokumentation der Abweichungen, Ebene A, interne Dokumentation der Abweichungen
- Wir sprechen erst dann von einer Methodik, wenn der Testprozess den oben genannten Ebenen entspricht.
- Testprozessmanagement, Ebene B, Planung, Durchführung, Überwachung und Anpassung
  Die Methodik darf nicht nur beschrieben sein, sondern ist auch einzuhalten. Zu diesem Zweck muss eine Überwachung und Anpassung des Prozesses stattfinden.

Optimierungsvorschläge

Legen Sie eine Beschreibung der Methodik in einem Handbuch oder einem Testplan fest, oder verweisen Sie auf Literatur. Dieser Schritt wird sich zum Teil aus der Sammlung von bestehendem Material zusammensetzen.

Sorgen Sie dafür, dass ausreichende Testsachkenntnis vorhanden oder zu erwarten ist (Ausbildungen), um zu gewährleisten, dass tatsächlich nach der beschriebenen Methodik vorgegangen werden kann.

13.B Projektspezifisch mit externem Betrachtungsbereich

Beschreibung

Der Betrachtungsbereich eines Projekts besteht nicht nur aus der Arbeit, die innerhalb des Testteams geleistet wird, sondern auch aus allen Schnittstellen mit Lieferanten und/oder Auftraggeber, sowohl intern als auch extern. Diese Schnittstellen müssen formalisiert werden, um die Informationen zu liefern, die erforderlich sind, um die in der Automobilindustrie geforderte Qualität zu erreichen.

Die Methodik ist nur dann ausreichend, wenn für andere Kernbereiche ein gewisser Reifegrad erreicht ist. Wenn die Kette von Lieferanten und Unterlieferanten mehr als zwei Organisationen umspannt, müssen die notwendigen Informationen für diejenige am Ende der Kette durch einen Vertrag oder Arbeitsvereinbarungen erzwungen werden. Ein Lieferant kann seine Schnittstellen mit seinem Vorlieferanten auf dieselbe Art und Weise Organisieren wie die Schnittstellen mit seinem Auftraggeber.

Kontrollpunkte

13.B.1 Die Abhängigkeiten die die Schnittstellen zwischen Auftraggeber und Lieferanten bilden, sind erfüllt worden: Teststrategie, Phasenmodel, Kommunikation, Berichterstattung, Dokumentation der Abweisungen und Testware Management.

Abhängigkeiten

- Testware Management, Ebene B, Externes Management von Testbasis und Testobjekt
- Dokumentation der Abweichungen, Ebene B, Umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungsmöglichkeiten
- Kommunikation, Ebene B, Projektkommunikation
- Berichterstattung, Ebene B, Fortschritt, Aktivitäten, Abweichungen mit Prioritäten
- Teststrategie, Ebene B, Kombinierte Strategie für High-Level Tests.
  Nur wenn der Testprozess die vorgenannten Ebenen erfüllt, sprechen wir von einer Methodik.

Optimierungsvorschläge

Erklären Sie Kommunikation, Berichterstattung, Management von Abweichungen und von Testware zu einem Standardbestandteil von Verträgen oder Arbeitsvereinbarungen zwischen Auftraggeber und Lieferant.

Verknüpfen Sie die Eingangs- und Ausgangskriterien mit geforderten Informationen wie z. B.: offene und behobene Abweichungen.

13.C Organisationsgenerisch

Beschreibung

Die Beschreibung einer Projektmethodik ist ein guter Ausgangspunkt für eine organisationsweite Einführung einer Standardmethodologie. Jedes neue Testprojekt kann dann diese Methodologie übernehmen. Da sich Testprojekte meist recht ähnlich sind, kann eine gemeinsame Toolsammlung verwendet werden, um den Testprozess zu unterstützen. Weitere Vorteile sind der Aufbau einer gemeinsamen Wissensbasis im Testen und ein erweitertes Verbesserungspotential, da mehr Personen dieselbe Methodologie einsetzen. Ein Vergleich zwischen Testprojekten kann den Grund dafür aufzeigen, warum das eine mit weniger Problemen zu kämpfen hat als das andere. Das Risiko einer organisationsweiten Methodologie besteht darin, dass entweder der Ansatz zu generisch ist, was dazu führt, dass jedes Testprojekt vieles selbst herausfinden muss, oder er ist zu detailliert, was nicht genügend Raum für die speziellen Bedürfnisse der einzelnen Testprojekte lässt.

Kontrollpunkte

13.C.1 Die Methodik ist in einem generischen Modell für die Organisation definiert.

13.C.2 Jedes Projekt arbeitet nach diesem generischen Modell.

13.C.3 Abweichungen sind hinreichend begründet und dokumentiert.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Erstellen Sie eine Beschreibung der organisationsweiten Methodologie in einem Handbuch oder Verweisen Sie auf entsprechende Literatur. Dieses Handbuch wird zum Teil daraus bestehen, dass man existierendes Material zusammenträgt. Widmen Sie dem Unterschied zwischen generischen und spezifischen Anteilen viel Aufmerksamkeit: Spezifische Punkte sollten nicht in der organisationsweiten Methodologie definiert werden.

Organisieren Sie das Testen in einer dedizierten Testabteilung, die dafür verantwortlich ist, die Methodologie zu etablieren, Personal in der Anwendung dieser Methodologie zu schulen und Testprojekte personell zu besetzen.

Sorgen Sie für strukturelle Kommunikation zwischen den Projekten und dieser Testabteilung (z. B.: um Abweichungen zwischen den Anweisungen zu diskutieren).

Beispiel:

Die Testorganisation kann als Linienabteilung organisiert werden, die die Möglichkeit bietet, einen standardisierten Testprozess zu implementieren. In dieser Form wird der Testprozess als Fabrik angesehen, die über Personal (Tester), Maschinen (Infrastruktur und Tools) etc. verfügt. Verschiedene Kunden (Projekte etc.) können Ihre Testaufgaben an diese Testorganisation auslagern. Der Auftraggeber kommt mit seiner Anforderung zu dieser Fabrik, Die Anforderungen werden in Form von Arbeitsaufträgen für das Personal eingeplant, die Infrastruktur wird korrekt angepasst, der Auftrag wird ausgeführt und der Auftraggeber kann das Produkt (Das getestete Produkt, mögliche Abweichungen Berichtswesen und Empfehlungen) zum vereinbarten Zeitpunkt entgegennehmen. Die Qualitätsstandards der Fabrik garantieren dem Auftraggeber eine gewisse Testqualität.

Das dieser Prozess in der Regel für jeden Auftrag derselbe ist und die Infrastruktur und das Personal bereits in der Fabrik verfügbar sind, ist es weitaus effizienter und die Tests können in deutlich kürzerer Zeit durchgeführt werden als wenn der Testprozess jedes Mal von Grund auf (für jedes Projekt) neu aufgesetzt werden müsste. Eine Voraussetzung für einen solchen Prozess ist, dass der Auftrag mit der verfügbaren Infrastruktur abgewickelt werden kann. Wenn eine neue Testinfrastruktur bestellt werden muss, benötigt man „Rüstzeit". Obwohl dieser Prozess das Testen schneller, billiger und besser macht, sind einige Punkte von größerer Bedeutung als gewöhnlich:

- Feste Vereinbarungen und gute Kommunikation zwischen dem Auftragnehmer und dem Auftraggeber
- Genügend technische Ausstattung,
- Verfügbarkeit von Fachkenntnis.

13.D Organisationsoptimierend, F&E Aktivitäten

Beschreibung

Eine generische Vorgehensweise darf nicht „statisch" sein, sondern muss überwacht und angepasst werden, um den sich ändernden Verhältnissen zu entsprechen. Bei Entwicklungen, die neu für eine Organisation sind (beispielsweise: „Rapid Application Development"), muss ermittelt werden, ob das Testkonzept anzupassen ist. Ferner wird versucht, auf dieser Ebene das Optimum zwischen generischen und spezifischen Anteilen so gut wie möglich zu erreichen.

Kontrollpunkte

13.D.1 Es findet ein strukturierter Feedback Prozess (sowohl formal festgeschrieben als auch durch die F&E Abteilung implementiert) zum generischen Modell statt.

13.D.2 Das generischen Modell wird strukturell gewartet und erneuert (durch F&E), u.a. auf der Grundlage von Feedback.

Abhängigkeiten

- Engagement und Motivation, Ebene B, Testen in Projektorganisation integriert
  F&E auf dem Testgebiet erfordert ein hohes Engagement aller Beteiligten.
- Testprozessmanagement, Ebene C, Überwachung und Anpassung in der Organisation
  Eine Voraussetzung für die Aktualisierung und Innovation des generischen Modells ist, dass dieses Modell auch tatsächlich (in richtiger Weise) eingesetzt wird. Das erfordert eine entsprechende Überwachung.

Optimierungsvorschläge

Sorgen Sie dafür, dass die Linienabteilung Testen Überprüfungs- und Abschlussbesprechungen mit Mitarbeitern aus jedem Testprojekt führt, in denen das organisationsweite Testmodell erörtert und bewertet wird.

Stellen Sie regelmäßig, beispielsweise (halb)jährlich, einen Plan für die aufzunehmenden Punkte für Aktualisierung und Innovation auf. Beanspruchen Sie entsprechende Ressourcen für die Ausführung dieses Plans, und überwachen Sie die Ausführung.

Sorgen Sie dafür, dass die Produkte der Aktualisierung und Innovation im generischen Modell verarbeitet werden, und setzen Sie die Organisation von den Anpassungen in Kenntnis.

14 KOMMUNIKATION

In einem Testprozess muss auf alle möglichen Arten eine Kommunikation mit den verschiedenen Beteiligten stattfinden, sowohl innerhalb des Testteams als auch mit anderen Parteien, beispielsweise dem Entwickler, dem Anwender und dem Auftraggeber. Diese Kommunikationsformen sind für einen gut verlaufenden Testprozess von wesentlicher Bedeutung, sowohl zur Abstimmung der richtigen Voraussetzungen und der Teststrategie als auch zur Kommunikation über den Fortschritt und die Qualität.

14.A Interne Kommunikation

Beschreibung

Eine gute Kommunikation zwischen den Teilnehmern am Testteam ist sowohl für ein gutes gegenseitiges Einvernehmen als auch für Motivation und gegenseitiges Verständnis wichtig. Kommunikation führt auch zu einer früheren Identifizierung von (anstehenden) Problemen, so dass entsprechende Maßnahmen rechtzeitig getroffen werden können. In kleinen Teams ist die Kommunikation meist in recht informeller Weise organisiert. Auch in solchen kleinen Teams ist es wichtig, dass getroffene Entscheidungen dokumentiert werden. Ein periodisches Treffen mit eher offiziellem Charakter ist notwendig, um ein Instrument zur Beurteilung der bisherigen Prozesse an der Hand zu haben. Die Frequenz dieser Meetings wird recht gering sein.

Kontrollpunkte

14.A.1 Innerhalb des Testteams und innerhalb des Entwicklungsteams finden regelmäßig Besprechungen statt. Diese Besprechungen haben eine feststehende Agenda und richten sich vornehmlich auf den Fortschritt (Durchlaufzeit und eingesetzte Arbeitsstunden) und die Qualität des zu testenden Objekts. Die Ergebnisse dieser Besprechung werden durch Notizen, Protokolle oder eine Statusliste dokumentiert.

14.A.2 Jedes Teammitglied nimmt regelmäßig an diesen Besprechungen teil.

14.A.3 Abweichungen vom Testplan werden mitgeteilt und schriftlich festgehalten.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Bitten Sie jedes Teammitglied regelmäßig um eine Bewertung des Testprozesses: Was läuft gut, und was könnte besser laufen?

Sorgen Sie für eine konsistente Bearbeitung der Aktionen, die sich aus den Besprechungen ergeben.

Sorgen Sie dafür, dass Projektneuigkeiten in den Besprechungen mitgeteilt werden.

14.B Projektkommunikation (Abweichungen, Änderungsüberwachung)

Beschreibung

Projekte sind selten von ihrer Umgebung losgelöst. Ein OEM-Projekt hat mit internen und externen Lieferanten zu tun und ein Lieferant hat oft mit Vorlieferanten und zumindest einem Auftraggeber zu tun. Selbst innerhalb eines einzelnen OEM gibt es Fälle, in denen eine Abteilung ein Lieferant für eine andere Abteilung ist. Daher ist es notwendig, Kommunikationsstrukturen aufzubauen, die die Grenzen eines Projektteams überschreiten. Der Lieferant muss zeitnah über Änderungen der Anforderungen informiert werden. Zeitnah bedeutet, dass der Lieferant in der Lage ist, die Änderungen zu implementieren, aber auch, die Änderungen hinreichend zu testen, wie es mit dem Auftraggeber vereinbart ist. Bei einer Teillieferung (A, B Muster, etc.) muss es für den Auftraggeber klar sein, welche Funktionalität eingeschlossen ist, welche offenen Abweichungen noch bestehen und welche Tests durchgeführt werden.

Eine periodische Besprechung über Abweichungen zwischen Auftraggeber und Lieferant ist ein nützliches Instrument, um Informationen über Fortschritt, bisherige Qualität des Produkts und mögliche Engpässe, z. B. Verfügbarkeit von Testausstattung. In diesen Besprechungen können offene Abweichungen priorisiert werden und es kann geplant werden, in welcher Lieferungsabfolge die Lösungen implementiert werden sollten.

Sowohl Auftraggeber als auch Lieferant haben ihre eigenen Verantwortlichkeiten in diesem Prozess. Der Auftraggeber muss die richtigen Informationen über Fortschritt, Qualität und Verfügbarkeit von Funktionalitäten einfordern. Diese Information ist notwendig, um die Integration mit Komponenten zu planen, die von anderen Lieferanten beigestellt werden. Der Lieferant muss die Eingangskriterien für die in Auftrag gegebene Systemkomponente, die Verfügbarkeit von Testausstattung und den Lieferplan für die Funktionalitäten abfragen. Weitere Informationen, die man erfragen sollte, sind wann die Zielhardware verfügbar sein wird und falls dies verspätet geschieht, ob Simulationsumgebungen oder Modelle verfügbar sind. Hier ist eine enge Zusammenarbeit zwischen Auftraggeber und Lieferant erforderlich, um sicherzustellen, dass hinreichendes Testen möglich und erschwinglich ist und dass dies im Zeitplan geschieht.

Änderungen der Anforderungen müssen auf ihre Auswirkungen für Konzeption und Testen überprüft werden. Das letztere wird oft vergessen. Kleine Designänderungen können wesentliche Auswirkungen auf das Testen haben (z. B. wegen Abhängigkeiten oder Regression). Wenn Änderungen der Anforderungen erforderlich werden, muss daher der Tester informiert und nach seiner Einschätzung der Auswirkungen auf das Testen gefragt werden. Dann ist zumindest ein Anhaltspunkt für die zusätzliche Zeit, das Geld und die Testressourcen in Relation zu den Änderungen der Anforderungen verfügbar.

Kleine Teams müssen eine Balance finden zwischen einer möglichst geringen Zeitverschwendung durch Besprechungen und der Notwendigkeit, Entscheidungen und Änderungen zu dokumentieren, um das Projekt Transparent, kontrollierbar und wartbar zu halten. Daher erscheinen die Kontrollpunkte 14.B.1 bis 14.B.4 sehr formal, aber ohne eine minimale Implementierung (Dokumentation von Fortschritt und Entscheidungen und Nachhalten der enthaltenen Risiken) dieser Punkte läuft das Projekt langfristig aus dem Ruder.

Kontrollpunkte

14.B.1 Die internen Testbesprechungen werden schriftlich protokolliert.

14.B.2 Bei den internen Testbesprechungen ist, neben dem Fortschritt und der Qualität des Testobjekts, auch die Qualität des Testprozesses ein fester Tagesordnungspunkt.

14.B.3 Der Testmanager berichtet regelmäßig während der Projektbesprechung über den Fortschritt und die Qualität des zu testenden Objekts, einschließlich der Risiken. Der Testmanager berichtet ebenfalls über die Qualität des Testprozesses.

14.B.4 Vereinbarungen, die bei diesen Treffen festgelegt werden, werden dokumentiert.

14.B.5 Der Testmanager wird unverzüglich über Änderungen im geplanten und vereinbarten Freigabedatum sowohl der Testbasis als des Testobjekts und der Testumgebung (z. B. Mechanische Komponenten, Prototypfahrzeuge, Simulatoren, Software, Modelle) in Kenntnis gesetzt.

14.B.6 Bei einer in regelmäßigen Abständen stattfindenden Auswertungsbesprechung (auch „Analyseforum" genannt) werden Testergebnisse zwischen den Vertretern des Testteams und anderen beteiligten Gruppen (z. B. Lieferant und/oder Auftraggeber) erörtert.

14.B.7 Tester sind beim „change control" an der Beurteilung der Auswirkung von Änderungsvorschlägen im Testaufwand beteiligt.

14.B.8 Vereinbarungen zur Unterstützung werden zwischen Testteam und dem Lieferanten des Testobjekts getroffen. Diese Vereinbarungen beinhalten: Beheben von Abweichungen, die den Testfortschritt blockieren, Kommunikationswege und Eskalationsverfahren

Abhängigkeiten

- Berichterstattung, Ebene B, Fortschritt, Aktivitäten, Abweichungen mit Prioritäten
- Phasenmodell, Ebene A, Planung, Spezifikation und Durchführung
  Die Teilnahme an den Projektbesprechungen, in denen das Testteam über die Qualität und den Fortschritt berichtet, bedeutet, dass der Testprozess überprüfbar sein muss. Der Einsatz eines Phasenmodells ist dabei besonders wichtig. Ferner ist auf einer bestimmten Ebene Bericht zu erstatten.
- Dokumentation der Abweichungen, Ebene A, interne Dokumentation der Abweichungen
  Voraussetzung für eine Teilnahme an der Auswertungsbesprechung ist die korrekte interne Verwaltung der Fehler und Abweichungen.

Optimierungsvorschläge

Sorgen Sie dafür, dass der Test in der Projektbesprechung von jemandem vertreten wird, der nicht noch zusätzlich für andere Aktivitäten (insbesondere Entwicklungsaktivitäten) verantwortlich ist. Ansonsten besteht das Risiko, dass vom Testen ausgehende Signale zu sehr abgeschwächt werden.

Stellen Sie ein Verfahren für die periodisch stattfindenden Auswertungsbesprechungen auf. Vergessen Sie dabei nicht die Möglichkeit einer Eskalation und eines Eskalationsverfahrens (bei Fehlern, die den Testfortschritt blockieren). Es ist besonders wichtig, dass in dieser Besprechung sowohl Vertreter des Auftraggebers als auch des Lieferanten anwesend sind. Diese Vertreter müssen mit der Autorität ausgestattet sein, über Prioritäten, Änderungen von Lieferungen und geplante Lieferdaten zu entscheiden.

Beginnen Sie mit den Auswertungsbesprechungen, widmen Sie dabei dem Lerneffekt der Formulierung von Fehlern entsprechende Aufmerksamkeit (Tester schreibt nur: „Berechnung stimmt nicht", Programmierer erwartet hier: In Zeile 124 in Programm 23a steht ein „-" anstatt eines „+"; das muss zu einer Vereinbarung über eine gute Detaillierung von Abweichungen führen). Deutliche Fehlerbeschreibungen ersparen eine Menge Sucharbeit und Kommunikation.

Berücksichtigen Sie bei der Veranschlagung der Änderungen das Regressionstesten. Das gleichzeitige Testen von einigen Änderungen ist sehr viel kostengünstiger als nacheinander durchgeführte Einzeltests (und der Übergang in eine nächste Systementwicklungsphase) dieser Änderungen. Das ist übrigens ein wichtiges Argument, um (Gruppen) von Änderungen in Releases auszuliefern. Sorgen Sie dafür, dass bei der Veranschlagung von Änderungen die oben genannten Aspekte berücksichtigt werden; dabei sind Kenntnisse über eine fundierte Veranschlagung von Tests erforderlich.

Ermöglichen Sie, dass entweder der Test- oder der Projektmanager die zusammenfassende Testberichterstattung in der Lenkungsgruppe bekannt macht.

14.C Kommunikation über die Qualität der Testprozesse auf Organisationsebene

Beschreibung

Die Verbesserung von Testprozessen ist eine andauernde Aktivität. Die Organisation ist sich darüber im Klaren, dass ein gut eingerichteter Testprozess viel zur Überwachung der Qualität und der Kosten des Produkts beiträgt. Die Einrichtung von Besprechungen auf Organisationsebene sorgt dafür, dass die vorhandenen Kenntnisse zu allen betroffenen Personen gelangen und auf diese Weise in der Organisation verfügbar sind. Das vereinfacht die Einrichtung neuer Testprozesse (mit neuen Mitarbeitern).

Kontrollpunkte

14.C.1 Besprechungen, in denen Vorschläge zur Verbesserung der eingesetzten Testmethodik und der Testprozesse besprochen werden, finden regelmäßig statt.

14.C.2 Teilnehmer sind Vertreter des Testteams und der Linienabteilung Testen.

Abhängigkeiten

- Reichweite der Methodik, Ebene C, Organisationsgenerisch
  Kommunikation auf Organisationsebene über das Testen ist wenig sinnvoll, wenn jeder Testprozess nach einem anderen Konzept stattfindet. Eine generische Methodik ist daher Voraussetzung.

Optimierungsvorschläge

Ernennen Sie jemanden aus der für das Testen verantwortlichen Linienabteilung als Organisator für die (periodischen) Besprechungen.

Stellen Sie eine feste Agenda auf, mit Aktionspunkten usw.

Beziehen Sie die Tester der verschiedenen Teststufen in die Besprechungen mit ein (auch die Vertreter der Low-Level Tests).

Beziehen Sie die Entwickler ad hoc in die Besprechungen mit ein.

Überwachen Sie die korrekte Bearbeitung von Hinweisen und Optimierungsvorschlägen aus den verschiedenen Testprozessen. Es ist nicht schlimm, wenn ein Vorschlag nicht angenommen wird, wohl jedoch, wenn nie mehr darauf zurückgekommen wird.

Sorgen Sie insbesondere am Anfang dafür, dass bereits einige Optimierungsvorschläge besprochen werden und dass der Fortschritt bei laufenden Verbesserungen angegeben werden kann.

Beziehen Sie die Lieferanten in die Optimierung des Testprozesses ein, indem sie zu bestimmten Zeitpunkten während des Projekts die Kooperation bewerten. Wenn der Lieferant ein so genannter Hauslieferant ist, sind noch stärker formalisierte Prozeduren zum Testen möglich.

Wenn gleichzeitig „Software Process Improvement" Initiativen durchgeführt werden, empfiehlt es sich, einen dort beteiligten Mitarbeiter davon bei den Projektbesprechungen mit einzubeziehen, so dass beide Optimierungsaktivitäten weiterhin parallel verlaufen.

Kleine Gruppen können ein Netzwerk mit Testkollegen in anderen Abteilungen oder Unternehmen bilden, um gegenseitig Informationen auszutauschen und voneinander zu lernen.

15 BERICHTERSTATTUNG

Testen beschäftigt sich nicht nur mit dem Finden von Abweichungen, sondern hat auch zum Ziel, Einblick in die Qualität des Produkts und die verbleibenden Risiken zum Produktivstart zu erhalten. Daher ist die Berichterstattung das wesentlichste Produkt des Testprozesses. Sie sollte fundierte Ratschläge an den Auftraggeber zum Produkt und sogar zum Systementwicklungsprozess enthalten.

15.A Abweichungen

Beschreibung

Die erste Ebene beinhaltet, dass eine Berichterstattung überhaupt vorliegt. In dieser Berichterstattung werden zumindest die Anzahl der gefundenen und die Anzahl der noch offen stehenden Abweichungen angegeben. Das vermittelt einen ersten Eindruck der Qualität des zu testenden Systems. Wichtig ist außerdem eine periodische Berichterstattung.

Kontrollpunkte

15.A.1 Die gefundenen Abweichungen werden periodisch festgehalten, getrennt nach gelösten und noch offen stehenden Abweichungen.

15.A.1 Im Voraus werden Vereinbarungen getroffen, vorzugsweise im Testplan festgelegt, welche Aspekte beim Berichtswesen berücksichtigt werden sollen:

- Inhalt der Berichte
- Intervall der Berichterstellung (periodisch, auf Anforderung und ad hoc)
- Adressaten der Berichte
- Formell/informell

15.A.1 Neben dem Auftraggeber des Tests muss an andere Beteiligte berichtet werden. Hierzu zählen z. B. andere Betroffene, wie der Entwickler des Systems.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Dokumentieren Sie allgemein, wie viele Abweichungen in etwa aufgedeckt worden sind, ungeachtet der Tatsache, ob sie gelöst wurden oder nicht.

Dokumentieren Sie die noch offen stehenden Abweichungen. Dabei handelt es sich sowohl um die Abweichungen, die noch gelöst werden, als auch um solche, die nicht gelöst werden, auch wenn sie berechtigt sind (diese bezeichnet man mit „known errors"; unter Zeitdruck werden diese Abweichungen meistens nicht behoben.

Sorgen Sie dafür, dass die Behebung der Abweichungen in ein schlüssiges verwaltungsmäßiges Verfahren eingegliedert wird.

15.B Fortschritt einschließlich Prioritätenzuweisung und Berichterstattung über Zeitaufwand und Testfortschritt

Beschreibung

Die Testberichterstattung beinhaltet zusätzliche Informationen in Form der geplanten, der bislang verwendeten und der noch erforderlichen Budgets und Durchlaufzeiten. Diese Informationen sind relevant, da der Auftraggeber dadurch einen besseren Überblick über die Kosten des Testens und die Durchführbarkeit der (Gesamt-)Planung erhält. Zudem wird die Berichterstattung der Abweichungen mit Kategorien entsprechend ihrer Bedeutung (Schweregradkategorien) versehen. Zehn „kosmetische" Fehler sind wahrscheinlich weniger schwerwiegend als ein produktionsblockierender Fehler: Das erhöht den Überblick über die Qualität des getesteten Systems.

Kontrollpunkte

15.B.1 Die Abweichungen werden festgehalten und nach klaren und objektiven Richtlinien in Schweregradkategorien eingeteilt.

15.B.2 Der Fortschritt einer jeden Testaktivität wird periodisch und schriftlich dokumentiert, u.a.: Durchlaufzeit, aufgewandte Stunden, was ist spezifiziert, was ist getestet, was ist dabei korrekt und was nicht korrekt verlaufen, und was muss noch getestet werden.

Abhängigkeiten

- Testprozessmanagement, Ebene B, Planung, Durchführung, Überwachung und Anpassung
- Einsatz des Phasenmodells, Ebene A, Planung, Spezifikation und Durchführung
  Um den Fortschritt schriftlich festhalten zu können, muss dieser bekannt sein. Das bedeutet, dass der Testprozess verwaltet werden muss.
- Dokumentation der Abweichungen Ebene A, interne Dokumentation der Abweichungen
  Voraussetzung für eine Berichterstattung über die gefundenen Abweichungen bedeutet, dass die Fehler intern gut verwaltet werden müssen.

Optimierungsvorschläge

Machen Sie im Projekt klar, dass allein die Tatsache, dass keine offen stehenden Abweichungen mehr vorhanden sind, nicht heißt, dass nach Beendigung des Tests eine positive Empfehlung formuliert werden kann. Beispiel: Es wurde ein Fehler in Funktion A gefunden, der möglicherweise einen strukturellen Charakter trägt und ebenfalls in den Funktionen B bis Z enthalten ist. Wenn der Fehler in Funktion A behoben wurde, sagt das nichts über die Möglichkeit aus, dass sich der Fehler noch in den Funktionen B bis Z befindet. Die Empfehlung könnte lauten, zunächst diese Funktionen noch einmal zu testen, bevor das System freigegeben wird.

Konzentrieren Sie sich auf die wichtigsten Abweichungen.

Bei einer Berichterstattung über den Fortschritt wird sichtbar; was das Testen bringt und wie viel Zeit jede Aktivität etwa kostet. Das vergrößert den Überblick und das gegenseitige Verständnis.

Bei einer Berichterstattung, die über die Organisationsgrenzen hinausreicht, kann ein Prinzip einer einzigen Quelle eingeführt werden.

15.C Risiken und Empfehlungen anhand von Metriken

Beschreibung

Die Risiken werden soweit wie möglich mit Trendanalysen von Metriken (Budgets, Zeit und Qualität (Abweichungen)) im Zusammenhang mit dem getesteten Objekt oder seinen Elementen belegt. Bei Risiken kann es sich um das Verstreichen der Frist handeln, zu der das Objekt eingesetzt werden soll, oder um eine unzureichende Qualität des getesteten Objekts. Für die festgestellten Risiken werden Empfehlungen formuliert, die insbesondere auf die Aktivitäten des Testens zielen. Man denke bei solchen Empfehlungen beispielsweise an die Durchführung eines vollständigen Regressionstests von Teilsystem A und eines beschränkten Regressionstests für Teilsystem B. Der wesentliche Vorteil ist, dass eine solche Berichterstattung den Auftraggeber in die Lage versetzt, rechtzeitig entsprechende Maßnahmen zu treffen. Die Untermauerung mit Trendanalysen liefert dem Auftraggeber die entsprechenden Argumente, um die (häufig kostenträchtigen) Maßnahmen zu treffen.

Kontrollpunkte

15.C.1 Über das Testobjekt wird ein Qualitätsurteil abgegeben. Dieses Urteil gründet sich auf die Akzeptanzkriterien, Eingangs- oder Ausgangskriterien - falls vorhanden - und wird auf die Teststrategie bezogen.

15.C.2 Regelmäßig werden mögliche Trends im Zusammenhang mit dem Fortschritt und der Qualität dokumentiert und es wird darüber berichtet.

15.C.3 Der Bericht enthält Aussagen zu Risiken (für den Auftraggeber bzw. für die Betriebsführung) sowie Empfehlungen.

15.C.4 Grundlage des Qualitätsurteils sowie der festgestellten Trends sind Statistiken (anhand der Dokumentation der Abweichungen und der Fortschrittsüberwachung), z. B. Anzahl Vergleiche der gefundenen Abweichungen gegenüber den durchgeführten Testfällen pro Zeiteinheit oder die durchgeführten Testfälle gegenüber den geplanten Testfällen.

Abhängigkeiten

- Metriken, Ebene A, Projektmetriken (über Produkt)
- Dokumentation der Abweichungen, Ebene B, umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungsmöglichkeiten
  Die für die Berichterstattung erforderliche Untermauerung mit Metriken erfordert mindestens diese Ebenen.
- Teststrategie, Ebene A, Strategie für einzelnen High-Level Test
- Testspezifikationstechniken, Ebene B, formale Techniken
  Zur Formulierung eines fundierten Urteils über das zu testende System sind Informationen über die Risiken notwendig. Zudem muss die Teststrategie auf diesen Risiken basieren.

Optimierungsvorschläge

Nehmen Sie die gewählte Teststrategie als Ausgangspunkt. Wurde davon abgewichen? War diese Strategie bereits „dünn"? Ist der Regressionstest eigentlich strukturiert verlaufen? Wie groß ist die Wahrscheinlichkeit einer Regression?

Stellen Sie diese Fragen bei jedem zu testenden Qualitätsmerkmal. Versuchen Sie, durch die Antworten hierauf eine Einschätzung der Risiken vorzunehmen, und schlagen Sie Maßnahmen vor.

Untermauern Sie die wichtigsten Schlussfolgerungen soviel wie möglich mit Tatsachen, d.h. mit Metriken aus der Fortschrittsüberwachung und Dokumentation der Abweichungen.

Ein Beispiel für das Inhaltsverzeichnis eines Fortschrittsberichts:

1. Einleitung
2. Vereinbarungen
3 Ausgeführte Aktivitäten
   - 3.1 Fortschrittsübersicht
   - 3.2 Trends, Anmerkungen und Empfehlungen zum Fortschritt

4 Qualität
   - 4.1 Qualitätshinweise und -charakteristiken
   - 4.2 Trends, Anmerkungen und Empfehlungen über die Qualität

5 Engpässe und Diskussionspunkte
6 Aktivitäten im kommenden Zeitraum

Das Inhaltsverzeichnis eines Abschlußberichts ist üblicherweise ähnlich dem eines Fortschrittsberichts:

1. (Management Zusammenfassung)
2. Einleitung
3. Vereinbarungen
4. Evaluation des Testobjekts
   - 4.1 Allgemein
   - 4.2 Freigabeempfehlung

5. Evaluation des Testprozesses
   - 5.1 Allgemein
   - 5.2 Empfehlungen für zukünftige (Test-)Projekte

6. Durchgeführte Aktivitäten
   - 6.1 Fortschrittsüberblick
   - 6.2 Trends, Anmerkungen und Empfehlungen zum Fortschritt

7. Qualität
   - 7.1 Qualitätsindikatoren und Charakteristiken
   - 7.2 Trends, Anmerkungen und Empfehlungen über die Qualität

15.D Empfehlungen haben einen Software Process Improvement Charakter

Beschreibung

Bei dieser Form der Berichterstattung richten sich die Empfehlungen nicht mehr ausschließlich auf Testaktivitäten, sondern auch auf Aktivitäten außerhalb des Testens bzw. auf den gesamten Systementwicklungsprozess. Man denke in diesem Rahmen an Empfehlungen, das Fachkonzept (zusätzlich) zu überprüfen, eine Versionsverwaltung einzurichten oder in der Projektplanung die erforderliche Zeit für die Übertragung der Software zu berücksichtigen. Mit dieser Form der Berichterstattung zielt das Testen etwas mehr auf die Verbesserung des Prozesses als auf die des Produkts sowie auf die Vermeidung von Fehlern (oder auf jeden Fall darauf, Fehler so früh wie möglich zu finden).

Kontrollpunkte

15.D.1 Empfehlungen werden nicht nur auf dem Testgebiet formuliert, sondern auch im Bereich anderer Projektelemente.

Abhängigkeiten

- Teststrategie, Ebene C, Strategie für High-Level Tests sowie Low-Level Tests oder Prüfungsstufen
  Um Empfehlungen zu anderen Projektelementen formulieren zu können, ist es wichtig, dass alle Testprozesse eine konsistente Gesamtheit bilden und gut aufeinander abgestimmt sind. Die Empfehlungen sind erst dann wertvoll, wenn sie vor dem Hintergrund ausreichender Informationen über den gesamten Testprozess formuliert werden (die Empfehlungen müssen über das Testen hinausgehen).
- Engagement und Motivation, Ebene C, Testengineering
  Außerdem muss die Organisation ein hohes Maß an Engagement für den Testprozess aufweisen, um die Testempfehlungen für andere Projektelemente „ernst zunehmen".

Optimierungsvorschläge

Fangen Sie klein an, mit Empfehlungen, die nur für das Projekt gelten.

Beziehen Sie in einer späteren Phase die Linienabteilung mit ein, weil Software Process Improvement projektübergreifend ist (u.a. auch die Wartung).

Sorgen Sie dafür, dass die Linienabteilung die Empfehlungen und Maßnahmen koordiniert und überwacht.

16 DOKUMENTATION DER ABWEICHUNGEN

Obwohl das Fehlermanagement Aufgabe der Projektleitung und nicht der Tester ist, sind die Tester am meisten darin involviert. Gutes Fehlermanagement sollte den Lebenszyklus eines Fehlers (von der Erkennung eines Fehlers, über dessen Behebung bis zur Dokumentation der Fehlerbehebung) überwachen und nebenbei statistische Daten liefern. Diese Daten können genutzt werden um z.B. fundierte Qualitätsaussagen zu machen. Dokumentation der Abweichungen, Ebene A:

16.A Interne Dokumentation der Abweichungen

Beschreibung

Die Speicherung der Abweichungen in einer Dokumentation hilft einerseits, eine gute verwaltungsmäßige Bearbeitung und Überwachung zu ermöglichen und andererseits stellt sie eine Informationsquelle zur Qualität des Systems dar. Die Bearbeitung und Überwachung ist wichtig, denn nur so kann vermieden werden, dass Abweichungen unkorrigiert bleiben, ohne dass die richtigen Personen darüber entschieden haben. Das bedeutet beispielsweise, dass ein Entwickler niemals eine Abweichung als unberechtigt zurückweisen darf, ohne dass ein Dritter sie noch einmal überprüft.

Um die Qualität eines Systems einschätzen zu können, ist es nicht nur interessant zu wissen, dass alle offen stehenden Abweichungen behoben wurden, sondern auch deren Anzahl und Art.

Kontrollpunkte

16.A.1 Jede Abweichung und deren Status wird beobachtet. Mögliche Zustände sind:

- neu
- zugewiesen
- in Arbeit
- verschoben
- abgewiesen
- bereit für Nachtest
- Nachtest OK
- geschlossen

16.A.2 Folgende Aspekte der Abweichung werden erfasst:

- eindeutige Nummer
- Mitarbeiter, der die Abweichung entdeckt hat
- Datum
- Schweregrad
- Testobjekt mit Versionsbezeichnung
- Problembeschreibung
- Statusangabe

Abhängigkeiten

- Keine.

Optimierungsvorschläge

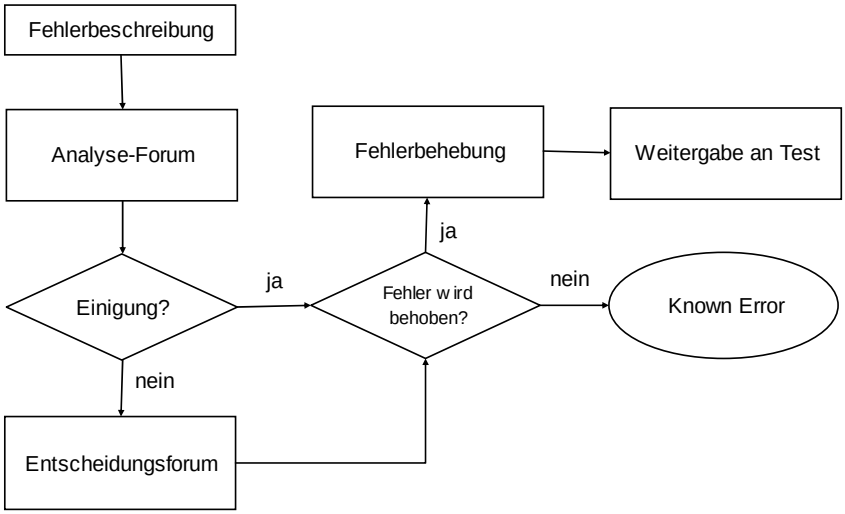
Das Führen einer solchen Verwaltung kann meistens mit einem Spreadsheet- oder Textverarbeitungsprogramm erfolgen, es sei denn, dass:

- eine sehr hohe Anzahl an Abweichungen erwartet wird (beispielsweise bei einem umfangreichen Projekt) bzw.
- umfangreiche Berichterstattungsmöglichkeiten gewünscht werden (siehe auch nächste Ebene).

Für diese Fälle ist es besser, ein spezifisches Tool für die Dokumentation der Abweichungen einzusetzen.

Führen Sie die Rolle des Vermittlers im Testteam oder Projekt ein. Diese Aufgabe soll die Abweichungsfindung und deren Lösungen adäquat kanalisieren. Der Vermittler unterhält dazu die externen Kontakte auf Managementebene. Der Vermittler fungiert als Mittelsmann zwischen Abweichungen auf der einen und deren Lösungen auf der anderen Seite. Vorteile dabei sind, dass die Qualität der Abweichungsfindung und der Lösungen besser überwacht und dass die Kommunikation gestrafft wird.

Ein Basisverfahren für die Abweichungsbehandlung ist in der folgenden Abbildung dargestellt:



Das Analyseforum ist eine Arbeitsplattform, an der u.a. Vertreter von Testern, Sachverständigen und Entwicklern teilnehmen. In diesen Besprechungen werden Entscheidungen über die Behandlung der Abweichungen getroffen. Wird man sich dabei nicht einig, oder hat eine Abweichung zu große Auswirkungen, greift man auf ein Entscheidungsforum zurück. In diesem Forum befinden sich Projektleiter und möglicherweise sogar Auftraggeber.

16.B Umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungsmöglichkeiten

Beschreibung

Von den Abweichungen werden verschiedene Daten festgehalten, die für eine gute Bearbeitung relevant sind. Auf diese Weise ist sowohl bei der Lösung als auch beim Regressionstesten deutlich, auf welchen Teil der Testbasis oder des Testobjekts sich die Abweichung bezieht und bei welchem Testfall die Abweichung derzeit aufgedeckt wurde. Durch umfangreiche Berichterstattungsmöglichkeiten können zusätzliche Informationen gesammelt werden, die dabei helfen, Trends so früh wie möglich zu signalisieren. Mögliche Trends sind beispielsweise die Feststellung, dass sich ein Großteil der Abweichungen auf einen Teil der Funktionsbeschreibungen bezieht oder dass die Abweichungen sich hauptsächlich auf einen speziellen Teil des Systems konzentrieren.

Diese Informationen können wiederum dazu verwendet werden, rechtzeitig einzugreifen und entsprechende Maßnahmen zu treffen.

Kontrollpunkte

16.B.1 Für die späteren Trendanalysen werden die Daten der Abweichung umfassend erfasst:

- Testfall
- Teststufe
- Systemteil
- Priorität (testblockierend Y/N)
- Testbasis + Version
- Ursache (vermutlich + definitiv)
- alle Statusübergänge der Abweichung, einschließlich Daten
- eine Beschreibung der Problemlösung
- (Version des) Testobjekts in der die Abweichung gelöst wird
- Problemlöser
- Testkonfiguration

16.B.2 Die Verwaltung eignet sich für umfassende Berichterstattungsmöglichkeiten; Übersichten können auf unterschiedliche Weise ausgewählt und sortiert werden.

16.B.3 Eine Person ist dafür zuständig, dass die Dokumentation der Abweichungen korrekt und konsequent durchgeführt wird.

16.B.4 Es findet eine Synchronisation zwischen den Systemen zur Dokumentation der Abweichungen auf Seiten des Lieferanten und auf Seiten des Auftraggebers statt. (z. B. Workflow, Status der Abweichungen, Attribute, Zeit für die Synchronisation) Dies bedeutet, dass zumindest offene Abweichungen, die vom Lieferanten entdeckt wurden, zum Zeitpunkt einer (Teil-)Lieferung in das Abweichungsmanagementsystem des Lieferanten eingetragen sein sollten.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Für eine solche Dokumentation der Abweichungen ist meistens eine automatisierte Unterstützung erforderlich (selbst entwickelt oder als kommerzielles Produkt erworben).

Geben Sie die Bedeutung einer Priorisierung bei Abweichungen an, um Diskussionen zu vereinfachen, um Verfahren schneller ablaufen zu lassen und mehr Einblick in die Testergebnisse zu erhalten. Besondere Aufmerksamkeit ist hier einer raschen Bearbeitung der Abweichungen zu schenken, die den Testfortschritt blockieren.

Treffen Sie Vereinbarungen über das Format, in dem die offenen Abweichungen kommuniziert werden. Dies macht es wesentlich einfacher und weniger fehleranfällig, die Abweichungen in das Abweichungsmanagementsystem der jeweils anderen Partei einzutragen.

16.C Dokumentation der Abweichungen wird im gesamten Projekt eingesetzt

Beschreibung

Eine einzige Dokumentation der Abweichungen für das gesamte Projekt bietet große Vorteile. Alle bei der Systementwicklung beteiligten Parteien, also u.a. die Entwickler; Anwender, Tester, Qualitätssicherungsmanager, können sowohl ihre gefundenen Abweichungen als auch eventuelle Lösungen erfassen und verfolgen. Die Kommunikation über die Bearbeitung der Abweichungen wird damit stark vereinfacht. Des Weiteren ist eine zentrale Verwaltung eine zusätzliche Quelle von Informationen.

Autorisierung und Sicherheit stellen einen wichtigen Aspekt dar, wenn mehr als eine Organisation involviert ist (was fast immer der Fall ist). Die Anzahl der Nachtests je Abweichung und die Anzahl der durch die Fehlerbehebung erzeugten Folgefehler gibt Aufschluss über den Reifegrad des Entwicklungsprozesses. Diese Art von Information muss für Personen außerhalb der Organisation unzugänglich sein. Ein Abweichungsmanagementsystem, das von mehr als einer Organisation benutzt wird, stellt ein mögliches Sicherheitsproblem dar. Personen außerhalb der Organisation benötigen einen Zugriff auf dieses System, meist über eine unsichere Internetverbindung. Daher sollten Sicherheitsvorkehrungen getroffen werden, um unautorisierten Zugriff auf dieses System und seine Informationen zu vermeiden.

Kontrollpunkte

16.C.1 Das Abweichungsmanagementsystem wird vom Auftraggeber (üblicherweise der OEM) bereitgestellt und ist für alle Parteien, die in das Projekt involviert sind, zugänglich (also auch für diejenigen außerhalb der Organisation des Auftraggebers).

16.C.2 Nur ein einziges System zur Dokumentation der Abweichungen wird im gesamten Projekt (z. B. die Entwicklung eines bestimmten Steuergeräts) eingesetzt, selbst dann, wenn mehr als eine unabhängige Organisation am Testprozess beteiligt ist. Die Abweichungen, die aus den unterschiedlichen Disziplinen, Teams (also Lieferant und Auftraggeber) oder Abteilungen gemeldet werden, werden im Abweichungsmanagementsystem zentral erfasst.

16.C.3 Jede Einheit (Auftraggeber, Lieferant, Vorlieferant) hat ihre eigene Ansicht auf das Abweichungsmanagementsystem. Diese Ansicht gewährt Zugriff auf genau die Art von Informationen, die notwendig sind, um ihre Tätigkeit auszuüben. Jede Einheit verwaltet ihre eigenen Informationen und entscheidet, welche Art von Informationen für wen zugänglich gemacht werden, indem Berechtigungsprofile eingesetzt werden.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Führen Sie Autorisierungsprofile je Einheit, die das Abweichungsmanagementsystem nutzt, ein. Die Autorisierung sollte durch die Einheit selbst überwacht werden. Jedes Autorisierungsprofil bezieht sich auf eine bestimmte Rolle. Die zugängliche Information beschränkt sich auf das absolute Minimum, das erforderlich ist, um diese Rolle auszufüllen. Mindestens zwei Autorisierungsprofile müssen je Einheit definiert werden, eines für internen Gebrauch und ein sogenanntes Gastprofil mit beschränktem Zugriff auf Informationen.

Eine zu eingeschränkte Dokumentation der Abweichungen reicht nicht aus, um im gesamten Projekt eingesetzt zu werden. Wenn die Dokumentation der Abweichungen

jedoch größtenteils (also nicht ganz!) Ebene B entspricht, kann es sinnvoll sein, die Dokumentation der Abweichungen im gesamten Projekt einzusetzen.

Legen Sie ein Verfahren für das periodische Abweichungsmeeting fest. Berücksichtigen Sie hierbei die Möglichkeit der Eskalation und eines ‚Kontingenzplans' (Im Fall von Abweichungen, die den weiteren Testfortschritt blockieren).

Aufgrund möglicher Sicherheitsprobleme ist es die beste Lösung, das Abweichungsmanagementsystem außerhalb des Unternehmensnetzwerks zu betreiben.

17 TESTWARE MANAGEMENT

Da es möglich sein muss, Testprodukte zu warten und zu übertragen, müssen sie verwaltet werden. Außer den Testprodukten selbst, wie Testplänen, -spezifikationen und -dateien, müssen auch die Produkte vorheriger Vorgänge wie Entwurf und Entwicklung gut verwaltet werden, da der Testprozess u.a. durch die Anwendung falscher Programmversionen schwer gestört werden kann. Wenn die Tester entsprechende Anforderungen an das Management stellen, wirkt sich dies positiv aus und die Testbarkeit der Produkte wird erhöht.

17.A Internes Testware Management

Beschreibung

Eine (gute) (Versions-)Verwaltung der internen Testware, beispielsweise der Testspezifikationen, -dateien und -datenbanken ist erforderlich, um die (Regressions-) Tests schnell durchführen zu können. Es kostet dann nicht so viel Zeit, um kurz vor der Testdurchführung zu ermitteln, was genau getestet werden muss und was wozu gehört. Außerdem werden Änderungen in der Testbasis zur Folge haben, dass auch Testfälle angepasst werden müssen. Um herauszufinden, um welche Testfälle es sich handelt, ist eine klar definierte und dokumentierte Beziehung zwischen Testbasis und Testfällen sehr wichtig.

Kontrollpunkte

17.A.1 Die Testware (Testfälle, Testskripte, (Ausgangs-)Datenbanken usw.), Testbasis, Testobjekt, Testumgebung (zusätzliche Komponenten - Hardware und Software), Testkonfiguration, Testdokumentation und Testvorschriften werden intern nach einem vorgeschriebenen Verfahren mit den entsprechenden Schritten für die Anlieferung, Erfassung, Archivierung und Benutzung verwaltet.

17.A.2 Das Management zeigt die Beziehungen zwischen den verschiedenen Elementen (u. a. Testbasis, Testobjekt, Testware) auf und kontrolliert sie.

17.A.3 Die Übergabe an das Testteam findet nach einem festgelegten Verfahren statt. Der Inhalt der Übergabe muss bekannt sein:

- Welche Komponenten und Versionen des Testobjekts (einschließlich Eigenentwicklungen, wiederverwendbare Komponenten und fertige Standardsoftware) werden übergeben.
- welche (Version der) Testbasis,
- gelöste Fehler, noch offen stehende Fehler, einschließlich der noch ungelösten Fehler des Entwicklers.
- Optional können andere Komponenten (z. B. Quellcode, benötigte Hard- und Software, Testware von vorigen Tests) Bestandteil der Übergabe sein.

Abhängigkeiten

- Keine

Optimierungsvorschläge

Betrauen Sie jemanden mit der Aufgabe des Testware Managements. Setzen Sie dazu vorzugsweise nicht den Testmanager ein, da dieser der „Auftraggeber" ist.

Legen Sie das Verfahren für das Management fest, und erläutern Sie dieses. Das folgende Beispiel verdeutlicht die Basisschritte:

Beispiel:

Die vier Schritte der Verfahrens 'Testware Management':

Lieferung

- Die zu verwaltenden Produkte werden von den Testern an den Testware Manager übergeben. Die Produkte müssen vollständig übergeben werden (Mit Datum und Versionsnummer). Der Manager überprüft die Vollständigkeit. Produkte in elektronischer Form sollte mit einer Standardnomenklatur übergeben werden, die auch die Versionsnummer enthält.

Registrierung

- Der Testwaremanager registriert die gelieferten Produkte in seiner Verwaltung mit Referenzen zu z. B. dem Lieferantennamen, Produktnamen, Datum und Versionsnummer. Wenn er geänderte Produkte registriert, sollte der Manager überwachen, dass die Konsistenz zwischen den unterschiedlichen Produkten gewährleistet bleibt.

Archivierung

- Man unterscheidet zwischen neuen und geänderten Produkten. Im Allgemeinen werden neue Produkte dem Archiv hinzugefügt und geänderte Produkte ersetzen die jeweils vorhergehende Version.

Referenz

- Die Ausgabe von Produkten an Projektteammitglieder oder an Dritte erfolgt durch eine Kopie der angeforderten Produkte. Der Manager erfasst, welche Version der Produkte wann an wen ausgegeben wurde.

Prüfen Sie die Möglichkeit, Tools zur Konfigurationsverwaltung und / oder zur Anforderungsverwaltung einzusetzen.

17.B Externe Verwaltung der Testbasis und des Testobjekts und Zuordenbarkeit von Systemanforderungen zu Testfällen

Beschreibung

Um Änderungen an Anforderungen, Testfällen und der Konfiguration des Testobjekts nachzuhalten, muss das Anforderungsmanagement zentralisiert werden. Dies eröffnet die Möglichkeit, Anforderungen direkt zu Testfällen und zum Testobjekt in Beziehung zu setzen. Ein kurzer Überblick darüber, was auf Basis welcher Anforderungen mit welcher Version des Testobjekts getestet wird, ist dann recht einfach. Änderungen an den Anforderungen lassen sich dann leicht den zugehörigen Testfällen zuordnen. Aus Änderungen am Testobjekt lassen sich direkt die Testfälle ableiten, die man braucht, um diese Änderungen zu testen.

Die Produkte der unterschiedlichen Phasen des Entwicklungszyklus stehen in gegenseitiger Beziehung zueinander: Aus Systemanforderungen lassen sich Softwareanforderungen ableiten, woraus sich wiederum das Softwaredesign ableiten lässt. Testfälle werden aus der Testbasis erstellt (die Systemanforderungen und/oder die Softwareanforderungen und/oder das Softwaredesign) und auf dem Testobjekt ausgeführt (Software, Benutzerhandbuch, etc.). Ein gutes Management dieser Beziehungen bietet für das Testen mehrere Vorteile:

- Es bietet viel Einblick in die Qualität und Tiefe des Tests, da dokumentiert ist, mit welchen Testfällen die Systemanforderungen, die Softwareanforderungen, das

Softwaredesign und die Software selbst getestet wurden (oder werden). Dies reduziert das Risiko von Lücken im Test.

- Im Falle von Änderungen an der Testbasis oder am Testobjekt lässt sich leicht zurückverfolgen, welche Testfälle angepasst oder erneut ausgeführt werden müssen.
- Wenn es als Folge hohen Zeitdrucks nicht möglich ist, alle geplanten Testfälle auszuführen, müssen Testfälle gestrichen werden. Da die Beziehungen zwischen Anforderungen, Spezifikationen und der Software bekannt sind, können diejenigen Testfälle gestrichen werden, für die die zugrunde liegende Anforderung oder die Spezifikationen das geringste operative Risiko beinhaltet, und es ist klar, für welche Anforderungen oder Spezifikationen weniger fundierte Aussagen über die Qualität gemacht werden.

Kontrollpunkte

17.B.1 Die Testbasis und das Testobjekt (meistens Entwurf und Software) werden vom Projekt nach einem vorgeschriebenen Verfahren mit den entsprechenden Schritten für die Anlieferung, Erfassung, Archivierung und Benutzung verwaltet.

17.B.2 Das Management umfasst die Beziehungen zwischen den verschiedenen Elementen (Testbasis, Testobjekt).

17.B.3 Das Testteam wird rechtzeitig über Änderungen in der Testbasis oder im Testobjekt informiert. Dies gilt auch für Anpassungen an kommerzieller Standardsoftware oder selbstentwickelte Softwarekomponenten.

17.B.4 Jede Anforderung und/oder Designobjekt steht in Beziehung zu mindestens einem Testfall.

17.B.5 Diese Beziehungen lassen sich über mehrere Versionen hinweg nachvollziehen (z. B. Systemanforderung A, Version 1.0, bezieht sich auf Funktionales Design B, Version 1.3, bezieht sich auf Programme C und D, Version 2.5 und 2.7 und bezieht sich auf Testfälle X bis Z, Version 1.4).

17.B.6 Kriterien für die Konformität mit den Systemanforderungen, Softwareanforderungen und Softwaredesign sind definiert und beziehen sich auf die richtigen Versionen dieser Anforderungen und dieses Designs (Es ist abhängig vom Umfang der Analyse, ob alle drei realisiert werden müssen.)

17.B.7 Die neuen Konfigurationseinstellungen (von internen oder externen Parteien bereitgestellt) werden nur in einem Standardformat ausgeliefert und akzeptiert.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Versuchen Sie, einige Beispiele dazu zu sammeln, was infolge einer unvorschriftsmäßigen (externen) Versionsverwaltung falsch gelaufen ist. Verwenden Sie diese Beispiele, um innerhalb des Projekts sowie dem Management klarzumachen, wie wichtig eine Versionsverwaltung ist, sowohl vom Standpunkt des Tests als auch von dem des Projekts aus. Generell ist die Versionsverwaltung ohne ein Konfigurationsmanagementtool schwierig.

Wenn die Versionsverwaltung unzureichend geregelt ist, weisen Sie dann in den Testempfehlungen folgendermaßen auf die Risiken hin: „Das System, das wir getestet

haben, weist eine gute Qualität auf; wir haben aber keine Sicherheit darüber, dass es sich hierbei um die Version handelt, die in Produktion genommen werden soll, oder ob es sich um die Version handelt, die der Auftraggeber zu erhalten erwartet." Geben Sie außerdem an, wie schwierig der Testprozess infolge einer unzureichenden Versionsverwaltung war, da viel Ermittlungsarbeiten erforderlich waren bzw. viele unnötige Abweichungen gefunden wurden, verursacht durch falsche Versionszusammenstellung.

Integrieren Sie das Versionsmanagement der Anforderungen, Modele, Software, Hardware (auch Testsoftware und –hardware) und Testware. Dies bietet die Möglichkeit, alle notwendigen Konfigurationselemente für eine bestimmte Systemversion an einer Stelle zusammenzutragen. Besondere Aufmerksamkeit sollte man den Konfigurationselementen widmen, die außerhalb der Organisation realisiert werden. Diese Elemente müssen in einem standardisierten Format ausgeliefert werden, das die fehlerfreie Einführung dieser Elemente in das Konfigurationsmanagementtool unterstützt.

Durchleuchten Sie erforderlichenfalls den Prozess für die Versionsverwaltung innerhalb des Projekts, und formulieren Sie Empfehlungen.

Beziehen Sie nicht nur die Spezifikationen, sondern auch die Systemanforderungen in die Testbasis ein. Oft bedeutet dies, dass einige Untersuchungen durchgeführt werden müssen.

Die Nichtfunktionalen Qualitätsanforderungen sind oft nicht klar formuliert. Starten sie eine Diskussion, wie diese gemessen und beurteilt werden sollen.

Diese Stufe bezieht sich mehr auf das Projekt oder die Organisation als auf den Testprozess selbst. Seien Sie sich dessen bewusst.

Stellen Sie im Rahmen der Verwaltung der Testware gute Verknüpfungen zwischen Testfällen, der Testbasis und dem Testobjekt zur Verfügung. Dies erfordert eine gute Versionsverwaltung.

17.C Übertragbare Testware

Beschreibung

Indem die Testware wieder verwendbar gemacht wird, vermeidet man, dass die arbeitsintensive Spezifizierung von Testfällen in einer nächsten Projekt- oder Wartungsphase erneut stattfinden muss. Obgleich das sehr logisch klingt, stellt sich in der Praxis häufig heraus, dass es in der Stressperiode kurz vor Auslieferung oftmals nicht gelingt, die Testware aktuell zu halten, und dass es nach Beendigung des Testes meist nicht mehr erfolgt. Es ist jedoch fast unmöglich, die unvollständige und nicht aktualisierte Testware eines anderen wieder zu verwenden. Da die Wartung meistens nur einen beschränkten Teil der Testware wieder verwenden wird, empfiehlt es sich, diesen Teil sorgfältig zu übertragen. Gute Vereinbarungen beispielsweise, welche Testware vollständig und aktualisiert übertragen werden muss, helfen erheblich bei der Vermeidung einer erneuten Spezifizierung.

Kontrollpunkte

17.C.1 Die Testprodukte (oder eine vorher vereinbarte Auswahl davon) werden nach Beendigung des Testes vervollständigt (= vollständig und aktualisiert) und an die Wartungsorganisation übertragen; anschließend wird die Übertragung formal bestätigt.

17.C.2 Die übertragenen Testprodukte werden tatsächlich wieder verwendet.

Abhängigkeiten

- Testspezifikationstechniken, Ebene B, formale Techniken
  Um Testware übertragbar zu machen, muss die Arbeitsweise (wie die Testware zustande gekommen ist) für jeden verständlich sein. Das bedeutet, dass bestimmte Techniken angewandt werden müssen.

Optimierungsvorschläge

Erwägen Sie, die Verwaltung der Testware am Ende eines Projektes zentral bei einer Linienverwaltung Testen durchzuführen.

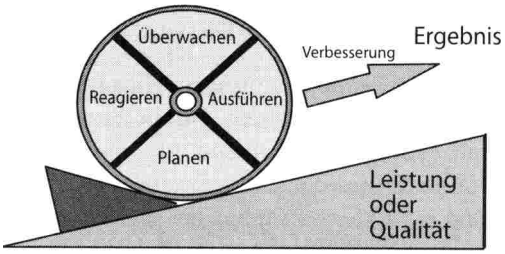
Das Problem bei der Aktualisierung der Testware besteht darin, dass relativ kleine Änderungen der Testbasis große Konsequenzen haben können. Stellen Sie sich vor, die funktionalen Spezifikationen werden innerhalb von 10 Minuten angepasst und der Programmierer implementiert die Änderungen in zwei Stunden, ist es in einer solchen Situation akzeptabel, dass der tatsächliche Test dieser Änderungen 4 Stunden benötigt und zusätzlich 20 Stunden benötigt werden, um die Testware anzupassen. Eine mögliche Lösung für dieses Dilemma besteht darin, dass man den Umfang der Testware reduziert, diese muss dann jedoch stets vollständig und aktuell sein. Diese Reduktion der Testware orientiert sich an den folgenden Erwartungen:

- Wie oft soll die Testware (wieder-)verwendet werden;
- Wieviel Zeit benötigt man jedes Mal, um eine Aktualisierung durchzuführen, verglichen mit der Zeit, die man benötigt, um alle Testfälle von Grund auf neu zu spezifizieren;
- Sollte die Testware auch später im Falle von Produkthaftungsfällen zur Verfügung stehen.

Aus diesem Grund müssen fundierte Vereinbarungen hierüber getroffen werden.

18 TESTPROZESSMANAGEMENT

Für die Überwachung eines jeden Prozesses und einer jeden Aktivität sind vier Schritte aus dem Deming Kreis [Deming, 1992] wichtig: Jede Aktivität wird geplant, ausgeführt, überwacht und erforderlichenfalls angepasst oder mit anderen Worten: „say what you do, do what you say".



Ein kontrollierter Testprozess ist äußerst wichtig, um in einem doch häufig turbulenten Testumfeld einen optimalen Test auszuführen.

Die Startebene ist durch das Fehlen einer Planung gekennzeichnet. Es wird sofort mit der Durchführung von Aktivitäten begonnen.

18.A Planung und Durchführung

Beschreibung

Ein wichtiger Schritt bei der Verbesserung des Testprozesses besteht darin, alle Aktivitäten im Vorhinein zu definieren und zu planen. Der Vorteil besteht darin, dass für jeden Beteiligten die Fragen „Was ist zu tun?", „Wann muss das fertig sein?" behandelt werden und damit klar wird, wie viel Zeit und Ressourcen eingeplant werden müssen. Diese Daten können im Rahmen des gesamten Projekts berücksichtigt werden. Ebene A dieses Kernbereichs ist sehr eng verbunden mit Ebene A des Kernbereichs „Einsatz des Phasenmodells".

Kontrollpunkte

18.A.1 Vor den eigentlichen Testaktivitäten wird ein Testplan aufgestellt, in dem alle auszuführenden Aktivitäten sowie die betroffenen Personen und ihre Verantwortlichkeiten identifiziert sind. Für jede Aktivität wird angegeben, in welchem Zeitraum diese stattfindet, welche Ressourcen (Personal oder Mittel) erforderlich sind und welches die zu liefernden Produkte sind.

18.A.2 Der Auftraggeber des Tests überprüft den Testplan, der aus der Planungsphase resultiert. Änderungen an diesem Plan müssen dem Auftraggeber zur Überprüfung vorgelegt werden.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Siehe Kernbereich „Einsatz des Phasenmodells", Ebene A, Planung, Design, Durchführung

Es empfiehlt sich, Aktivitäten mit einem geringen Umfang zu einer einzigen Aktivität zusammenzufassen.

18.B Planung, Durchführung, Überwachung und Anpassung

Beschreibung

Wenn Aktivitäten geplant werden, bietet das noch keine Garantie für eine (korrekte) Durchführung. Jeder Testprozess ist gekennzeichnet durch ein gewisses Maß an Chaos. Es gehört eher zur Regel als zur Ausnahme, dass im Laufe des Prozesses mit neuen Aktivitäten begonnen wird, Aktivitäten plötzlich nicht mehr relevant sind oder anders ausgeführt werden, als vorgeschrieben ist. Die Überwachung der (Ausführung der) Aktivitäten ist notwendig, um sicherzustellen, dass der Testprozess vollständig und sachgemäß durchgeführt wurde. Neben der Überwachung ist auch eine Anpassung erforderlich. Je nach den Projekteigenschaften und den erwarteten Risiken erfolgt dies entweder durch Anpassung der Planung oder durch Anpassung der Aktivitäten, die ausgeführt werden sollen.

Kontrollpunkte

18.B.1 Es findet eine Überwachung der Ausführung aller geplanten Aktivitäten statt.

18.B.2 Es findet außerdem eine zeitliche und finanzielle Überwachung einer jeden Aktivität statt.

18.B.3 Bei Abweichungen werden Anpassungen durchgeführt, entweder indem man die Planung anpasst, oder indem man erneut die Aktivitäten wie geplant durchführt. Die Anpassungen werden fundiert begründet.

18.B.4 Bei Abweichungen wird korrigierend eingegriffen, indem entweder die Pläne angepasst oder die Aktivitäten doch noch gemäß Plan ausgeführt werden. Der Eingriff wird begründet.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Änderungen und mögliche Anpassungen können in einer neuen Version des Testplans oder in Projektberichten festgelegt werden.

Sorgen Sie für eine Genehmigung der Abweichungen und der Anpassungen durch den Auftraggeber des Tests.

Sorgen Sie dafür, dass eine Gesamtübersicht über diese Abweichungen und Anpassungen am Ende des Testprojektes problemlos zu erstellen ist.

18.C Überwachung und Anpassung in der Organisation

Beschreibung

Auch auf der Organisationsebene findet eine Überwachung und Anpassung der verschiedenen Testprozesse statt, wenn auch auf einer viel allgemeineren Ebene. Es handelt sich hier vornehmlich um die Überwachung und Steuerung der Qualität der Testprozesse, wie z. B. den Einsatz der vorgeschriebenen Methodik (Methoden, Richtlinien, Techniken und Verfahren).

Kontrollpunkte

18.C.1 Auf Organisationsebene erfolgt eine Überwachung des Einsatzes der Methodik (Methoden, Richtlinien, Techniken und Verfahren) der Organisation.

18.C.2 Änderungen werden schriftlich festgelegt und den Beteiligten im Testprozess mitgeteilt.

18.C.3 Bei Abweichungen werden die Risiken analysiert, und es wird korrigierend eingegriffen, beispielsweise durch Anpassung der Methodik oder indem die Aktivitäten oder Produkte doch noch an die Methodik angepasst werden. Die Anpassungen werden begründet.

Abhängigkeiten

- Reichweite der Methodik, Ebene C , organisationsgenerisch
  Überwachung und Anpassung auf Organisationsebene ist nur möglich, wenn alle Testprozesse in der Organisation nach einer ähnlichen Arbeitsweise vorgehen.

Optimierungsvorschläge

Machen Sie die Linienabteilung Testen oder einen ihrer Vertreter für die Überwachung des Einsatzes der Methodik verantwortlich.

Sorgen Sie für Checklisten, auf deren Grundlage die Überwachung stattfindet.

19 PRÜFEN

Unter Prüfungen versteht man die Überprüfung von Zwischenprodukten, beispielsweise das Fachkonzept. Dies ist der wichtigste Unterschied zum Testen, das sich auf die Endprodukte bezieht. Prüfen wird oft als Teil der Qualitätssicherungsmaßnahmen organisiert und nicht als Teil der Testaktivitäten. Es ist daher nicht sicher, dass die Tester daran beteiligt sind. Im Vergleich zum Testen liegt der Vorteil von Prüfungen darin, dass Abweichungen im Entwicklungsprozess sehr viel eher entdeckt werden können. Die Korrekturkosten sind dadurch wesentlich niedriger. Außerdem kann eine Beurteilung sehr viel einfacher durchgeführt werden, da u.a. keine Programme laufen und keine Testumgebungen geschaffen werden müssen.

19.A Nichtformale Prüfungen

Beschreibung

Prüfungen, Kontrollen etc. erweisen sich immer und immer wieder als der effizienteste und effektivste Weg, Abweichungen in einem System (oder in seinen Zwischenprodukten) aufzudecken.

Kontrollpunkte

19.A.1 Checklisten werden in Kombination mit 4-Augen Überprüfungen für die Prüfung eingesetzt

19.A.2 Es findet eine Berichterstattung der Prüfungen und der gefundenen Ergebnisse statt.

19.A.3 Es erfolgt eine Überwachung der Bearbeitung der Ergebnisse.

19.A.4 Tester und Personen, die die unterschiedlichen Interessengruppen repräsentieren (z. B.: Projektleiter, Technische Experten, Entwickler und Auftraggeber) sind an diesen Prüfungen beteiligt.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Stellen Sie sicher, dass die berichteten Abweichungen aus den Prüfungen verwaltet werden. Die Lösung wichtiger Abweichungen sollte überprüft werden.

Führen sie 4-Augen Überprüfungen oder Kontrollen unter Kollegen mit dem Einsatz von Checklisten als Richtlinie ein. Neben den Checklisten nutzt der Partner oder Kollege seine eigene Erfahrung, um das Produkt zu inspizieren.

19.B Prüfungstechniken

Techniken, in Form von Prozessbeschreibungen und Checklisten helfen dabei, Prüfprozesse kontrollierbar zu gestalten. Es ist sehr wichtig, mehrere Techniken zur Auswahl zu haben, denn, ähnlich wie beim Testen muss nicht jedes Zwischenprodukt ebenso gründlich inspiziert werden, wie ein anderes.

Kontrollpunkte

19.B.1 Beim Prüfen von (Zwischen-)Produkten werden bestimmte Techniken verwendet. Es wird somit nach einer formalen und beschriebenen Arbeitsweise vorgegangen.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Sorgen Sie dafür, dass verschiedene Prüfungstechniken zur Verfügung stehen, aus denen bei der Bestimmung einer Prüfungsstrategie ausgewählt werden kann. Beispiele von Prüfungstechniken sind die Inspektionen, strukturierte Walkthroughs, 4-Augentest usw. Jede Technik hat im Großen und Ganzen folgende Phasen:

- Plan
  In dieser Phase erfolgt eine Identifikation der Risikobereiche, und es wird ermittelt, welche Analysemethoden am besten eingesetzt werden können (mit anderen Worten: Wie können wir vermeiden, dass nur auf Rechtschreib- und Stilfehler geachtet wird?). Die Ergebnisse werden in einem Review Plan festgelegt, zusammen mit den Antworten auf die Fragen nach dem Wer, Was und Wann und dem erforderlichen Aufwand.
- Analyse
  Eine Analyse kann durch Information, Untersuchung oder Diskussion stattfinden. Der Organisator erstellt eine Agenda, die zu untersuchenden Produkte werden analysiert und die gefundenen Probleme protokolliert. Größere Fehler werden verwaltet und überwacht; bei kleineren Fehlern wird dies häufig dem Autor überlassen.
- Korrektur
  Die Unstimmigkeiten werden bearbeitet. Bei bestimmten größeren Fehlern kann diskutiert werden, ob diese gelöst werden müssen; die Beseitigung manch kleinerer Fehler kann möglicherweise verschoben werden.
- Die Unterschiede zu der vorigen Version werden dokumentiert.
- Kontrolle
  Es erfolgt eine Kontrolle aller Änderungen, und es wird ein zusammenfassender Bericht erstellt.

Die meisten Prüfungstechniken basieren auf dem oben genannten generischen Phasenmodell, wobei sich die Variation vornehmlich in der angewandten Form der Analyse äußert:

- Walkthrough:
  Plan,
  Information,
  Korrektur,
  Kontrolle
- etc.

Versehen Sie jede eingesetzte Prüfungstechnik mit einem solchen (Mini-)Phasenmodell, in dem beschrieben ist, was wann von wem durchzuführen ist.

Ein Problem bei formalen Inspektionen ist, dass wiederholte Inspektionen häufig für zu teuer gehalten werden. Es muss vermieden werden, dass die Korrekturarbeiten dann überhaupt nicht kontrolliert werden. Es ist in einem solchen Fall besser, die Neuinspektion in vereinfachter Weise ausführen zu lassen.

19.C Prüfungsstrategie

Beschreibung

Genauso wie die Teststrategie ist eine Prüfungsstrategie sehr wichtig, einerseits, um den Aufwand optimal einzusetzen, und andererseits zur Kommunikation mit dem Auftraggeber.

Mit einer Strategiebestimmung wird analysiert, was, wo und wie intensiv geprüft werden muss, um das optimale Gleichgewicht zwischen dem gewünschten Einblick in die Qualität sowie der erforderlichen Zeit und dem benötigten Geld zu erhalten. Die Optimierung findet mit dem Ziel statt, die verfügbaren Ressourcen richtig über die auszuführenden Aktivitäten zu verteilen.

Kontrollpunkte

19.C.1 Es findet eine bewusste Risikoabwägung statt.

19.C.2 Es findet eine Differenzierung in Betrachtungsbereich und in der Tiefe der Prüfung statt, die von den eingegangenen Risiken und - falls vorhanden - von den Akzeptanzkriterien abhängig ist: Nicht alle Zwischenprodukte und Qualitätsmerkmale werden gleich intensiv getestet.

19.C.3 Aus mehreren Prüfungstechniken wird eine Auswahl getroffen, die sich für die gewünschte Tiefe der Prüfung eignet.

19.C.4 Auch bei erneuter Prüfung findet eine (einfache) Strategiebestimmung statt, bei der motiviert zwischen den Varianten „nur Lösungen prüfen" und „vollständig neu prüfen" gewählt wird.

19.C.5 Die Strategie wird aufgestellt und danach auch ausgeführt. Es wird überwacht, dass die Durchführung der Prüfungen gemäß Strategie stattfindet. Falls erforderlich wird steuernd eingegriffen.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Geben Sie die Risiken bei der gegenwärtigen Arbeitsweise an, oder weisen Sie darauf hin, dass die Prüfung schneller oder preiswerter durchgeführt werden kann.

Wenn nur eine Technik vorliegt, versuchen Sie dann, durch einfache Varianten mehr oder weniger Tiefe zu erzielen. Als Beispiel einer variablen Tiefe kann die (Nicht-) Einbeziehung bestimmter Personen genannt werden oder eine Einschränkung bzw. Ausdehnung der Anzahl der Fragen.

Stellen Sie für den erneuten Review eine Arbeitsweise auf, bei der jedes mal bewusst (und schriftlich festgelegt) die Abwägung zwischen vollständigem erneuten Prüfen oder „abgespecktem" erneuten Prüfen (je Fehler, Teilsystem oder Zwischenprodukt) gemacht werden muss.

Diskutieren Sie die verschiedenen Zwischenprodukte und Qualitätsmerkmale mit dem Auftraggeber, und versuchen Sie, die relative Bedeutung eines jeden Produkts oder Merkmals zu ermitteln.

Führen Sie schließlich eine vollständige Strategiebestimmung durch. Die zur Erstellung einer Teststrategie erforderlichen Schritte sind im Folgenden kurz beschrieben:

- Bestimmung der Qualitätsmerkmale
  Im Einvernehmen mit allen Beteiligten werden die Qualitätsmerkmale festgelegt, auf die sich die Prüfung konzentrieren soll. Während des Prüfprozesses erfolgt eine regelmäßige Berichterstattung über die ausgewählten Qualitätsmerkmale.
- Bestimmung der relativen Bedeutung der Qualitätsmerkmale
  Auf der Grundlage der Ergebnisse des vorigen Schrittes wird angegeben, wie der Prüfungsaufwand über die ausgewählten Qualitätsmerkmale zu verteilen ist, wobei zunächst davon ausgegangen wird, dass das Prüfen eines jeden Qualitätsmerkmals gleich arbeitsintensiv ist.
- Unterteilung Zwischenprodukte oder Teilsysteme
  Das System wird bei diesem Schritt in Zwischenprodukte und falls erforderlich, in weitere Teilsysteme unterteilt. Die Einteilung ist hierbei im Prinzip die gleiche wie bei der Entwurfsdokumentation. Wenn von der Einteilung abgewichen wird, ist dies deutlich zu begründen und zu beschreiben.
- Bestimmung der relativen Bedeutung der Zwischenprodukte oder Teilsysteme
  Auf der Grundlage der Ergebnisse des vorigen Schrittes wird angegeben, wie der Prüfungsaufwand über die unterschiedlichen Zwischenprodukte oder Teilsysteme verteilt werden muss, wobei auch hier zunächst davon ausgegangen wird, dass die Prüfung jedes Produkts gleich arbeitsintensiv ist. Geben Sie anschließend je Zwischenprodukt oder Teilsystem an, welche Qualitätsmerkmale relevant sind und wie intensiv diese, bezogen auf die zugewiesene Bedeutung, zu testen sind.
- Festlegung der einzusetzenden Prüfungstechniken
  Als letzter Schritt bei der Prüfungsstrategie werden Prüfungstechniken ausgewählt, anhand derer die im ersten Schritt bestimmten Qualitätsmerkmale und die festgestellten Teilsysteme geprüft werden.

20 LOW-LEVEL TESTS

Die Low-Level Tests werden von den Entwicklern durchgeführt. Bekannte Low-Level Tests sind der Unit Test und der Komponententest. Ähnlich wie bei der Evaluierung können Fehler durch diese Tests in einem früheren Stadium der Systementwicklung entdeckt werden als dies mit High-Level Tests möglich wäre. Low-Level Tests sind effizient, da sie wenig Absprachen erfordern, die Analyse einfacher ist und der Finder des Fehlers häufig auch derjenige ist, der ihn verursacht hat und ihn auch behebt.

Low-Level Tests ist die Teststufe, wo die Konformität mit Programmierstandards (wie z. B. der Programmierstandardteil von MISRA) gezeigt werden muss. Die Konformität mit einem solchen Standard kann dadurch sichergestellt werden, dass man die Entwicklungsumgebung um Code Checker ergänzt, was zudem die Möglichkeit bietet, Code Metriken über den Quellcode zu liefern (Dies wird von Auftraggebern zunehmend gefordert).

Dennoch ist die Qualität der durchgeführten Tests oft niedriger. Dies resultiert daraus, dass ein Entwickler im Vergleich zu einem Tester eine andere Grundeinstellung hat. Ein Entwickler möchte zeigen, dass das System fehlerfrei arbeitet, ein Tester versucht, den Unterschied zwischen der geforderten und der gelieferten Qualität aufzuzeigen, indem er konkret nach Abweichungen sucht.

20.A Phasenmodell für Low-Level Tests (Planung, Spezifikation und Durchführung)

Beschreibung

Der wesentliche Vorteil einer strukturierten Arbeitsweise bei Low-Level Tests liegt darin, dass der Prozess besser verwaltet werden kann und der Einblick in die Qualität des Tests größer wird. Wenn bei den Testaktivitäten nicht festgehalten wird, wann diese beginnen, wer den Test durchführt und was der Test beinhaltet, ist der Testprozess nicht verwaltbar. Man kann auf diese Weise keine Informationen über die Qualität des getesteten Objekts erhalten. Damit man dennoch bis zu einem gewissen Grad einen Einblick in die Qualität bekommt, müssen andere Teststufen dann mehr leisten.

Kontrollpunkte

20.A.1 Bei einem Low-Level Test werden (mindestens) folgende Phasen unterschieden: Planung, Spezifikation und Durchführung. Diese werden nacheinander ausgeführt, gegebenenfalls für jedes Teilsystem.

20.A.2 Die auszuführenden Aktivitäten für jede Phase werden im folgenden genannt. Jede Aktivität ist mit Unteraktivitäten bzw. -aspekten versehen. Diese verstehen sich als zusätzliche Informationen und haben daher keinen zwingenden Charakter.

- Für die Planungsphase:

Aktivität	Teilaktivitäten / Aspekte	Produkt
– Auftragsformulierung	– Auftraggeber und -nehmer – Betrachtungsbereich – Ziel – Randbedingungen	– Im Testplan oder Implementierungsplan festgelegt

	– Ausgangspunkte	
– Festlegung der Testbasis	– Bestimmung relevante Dokumentation – Bestimmung der erforderlichen Funktionen	– Im Testplan oder Implementierungsplan festgelegt
– Einrichten der Organisation	– Bestimmung der erforderlichen Funktionen – Zuweisung der Aufgaben, Befugnisse und Verantwortlichkeiten – Beschreibung der Organisation – Zuweisung des Personals	– Im Testplan oder Implementierungsplan festgelegt
– Beschreiben der Testprodukte	– Festlegung der Testprodukte – Erstellen (ggf. Auswählen) von Richtlinien	– Im Testplan oder Implementierungsplan festgelegt
– Definition Infrastruktur und Tools	– Definition Testumgebung – Definition Testtools	– Im Testplan oder Implementierungsplan festgelegt
– Definition des Testmanagements	– Definition Testprozessmanagement (Fortschritt, Qualität, Statistiken, Berichterstattung) – Definition Testproduktmanagement – Definition Bearbeitung der Abweichungen	– Im Testplan oder Implementierungsplan festgelegt
– Bestimmung der Planung	– Aufstellen der allgemeinen Planung	– Im Testplan oder Implementierungsplan festgelegt
– Festlegung des Testplans	– Bestimmung der Risiken, Gefahren und Maßnahmen – Festschreiben des	– Im Testplan oder Implementierungsplan festgelegt

	Testplans – Festlegung Testplan (Genehmigung Auftraggeber)	

- Für die Spezifikationsphase:

Aktivität	**Teilaktivitäten / Aspekte**	**Produkt**
– Aufstellen der Testspezifikationen und -skripte	– Testfälle (logisch und konkret) – Definition der Ausgangsdateien (Startbedingungen für die Ausführung von Testfällen) – Testskripte	– Testfälle – Definition Ausgangsdateien (Startbedingungen für die Ausführung von Testfällen) – Testskripte

- Für die Durchführungsphase:

Aktivität	**Teilaktivitäten / Aspekte**	**Produkt**
– Durchführung (Regressions-)Tests	– Durchführung Testskripte – Durchführung von statischen Tests (einschl. Beurteilung der Testergebnisse und Analyse der Abweichungen) – Erstellen von Codemetriken	– Dokumentation der Abweichungen – Testberichte

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Der rechtzeitige Beginn der Testvorbereitung für die Low-Level Tests muss so schnell wie möglich zur normalen Arbeitsweise werden.

Entwickler, Tester aber auch das Projektmanagement sollte sich dessen bewusst sein, dass der „Return on Investment" der Planung und der Testvorbereitung in kürzeren Durchlaufzeiten für die Testdurchführung und einer höheren Qualität der durchgeführten Tests besteht. Da die Testdurchführung üblicherweise auf dem kritischen Pfad liegt, wird auch die Durchlaufzeit des Gesamtprojekts kürzer. Auch muss man allen genannten Parteien bewusst machen, dass ein Low-Level Test von guter Qualität zu insgesamt niedrigeren Nacharbeitungskosten führt [Boehm, 1981]

Setzen Sie Testerfahrung ein, um die Low-Level Tests in Gang zu setzen.

Stellen Sie sicher, dass eine gewisse Unabhängigkeit zwischen dem Tester von Low-Level Tests und dem Entwickler besteht (z. B. Aufstellen der Testfälle der Funktion X durch eine andere Person als dem Entwickler dieser Funktion).

Die „Specialist Interest Group In Software Testing" (SIGIST) der „British Computer Society" (BCS) hat den 'Standard for Software Component Testing' bei der Britischen Normierungsbehörde „British Standards Institution" (BSI) eingereicht, um diese zu nationalen (und dann internationalen) Standards werden zu lassen. BS 7925-2, „Software component testing", spricht das Gebiet des Unit Tests an. Der Standard beschreibt einen generischen Testprozess, um ein konsistentes Referenzmodell innerhalb des Standards zu bieten. Es beschreibt eine Vielzahl von Testtechniken, die zur Auswahl stehen, hierbei sowohl „Black-Box" als auch „White-Box" Techniken. Sowohl die Testspezifikation als auch die hiermit verbundenen Testmaßtechniken werden beschrieben. Ein Anhang enthält Richtlinien, wie diese Techniken anzuwenden sind, wobei jede einzelne anhand von Beispielen beschrieben wird. [Auszug aus einer Web-Publikation von BCS SIGIST]

20.B White-Box Techniken

Beschreibung

Der Einsatz geeigneter Techniken für die Tests durch die Entwickler führt zu einer Steigerung der Qualität dieser Tests und damit der Qualität der Software. Von großer Bedeutung dabei ist, dass so viele Fehler wie möglich zu einem so früh als möglichen Zeitpunkt gefunden werden. Außerdem führt die strukturierte Arbeitsweise zu einer Vermeidung von Fehlern, da der Entwickler mehr Information darüber erhält, wo die Fehler gemacht worden sind. Schließlich bietet sie Möglichkeiten zur Prozessverbesserung, da bei einer beschriebenen und strukturierten Arbeitsweise besser sichtbar ist, wo und wie optimiert werden kann.

Kontrollpunkte

20.B.1 Neben nicht formalen Techniken werden bei den Low-Level Tests auch formale Testspezifikationstechniken angewandt, bei denen auf eindeutige Weise von der Testbasis zu den Testfällen gelangt werden kann.

20.B.2 Für Low-Level Tests ist es möglich, fundierte Aussagen über die Testabdeckung (bzgl. der Testbasis) zu treffen.

20.B.3 Die Testware ist durch die einheitliche Vorgehensweise (innerhalb des Testteams) wieder verwendbar.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Siehe die Anweisungen für den Kernbereich „Testspezifikationstechniken", Ebene B Formale Techniken.

Beginnen Sie mit nicht formalen Techniken, beispielsweise, indem Sie bei jeder zu testenden Bedingung auf einer Kopie der Funktionsspezifikation ankreuzen lassen, welche Bedingung getestet wurde.

Vermitteln Sie den Beteiligten die Bedeutung und die Vorteile von White-Box Testtechniken.

Verschaffen Sie sich Einsicht in Qualität und Tiefe der Low-Level Tests, indem Sie einfache Testtechniken einsetzen. Der Einsatz von schwierigen und formalen Techniken wird weniger leicht akzeptiert werden.

20.C Low-Level Teststrategie

Beschreibung

Der Auftraggeber eines Tests erwartet eine bestimmte Qualität des freizugebenden Systems. Die Qualitätserwartungen können für jedes System sehr unterschiedlich sein. Es ist von größter Bedeutung, mit dem Auftraggeber hierzu entsprechende Vereinbarungen zu treffen und diese anschließend durch die Art des Testens umzusetzen.

Eine Risikoeinschätzung bildet die Grundlage für die Teststrategie, da es wichtig ist, den Testaufwand zu optimieren. Während der Bestimmung der Teststrategie wird analysiert, was, wo und wieviel getestet werden muss, damit das erforderliche Gleichgewicht zwischen der gewünschten Qualität und dem zeitlichen oder finanziellen Aufwand, der dafür notwendig ist, gefunden wird. Eine Optimierung findet mit dem Ziel statt, die verfügbaren Ressourcen richtig über die auszuführenden Testaktivitäten zu verteilen.

Kontrollpunkte

20.C.1 Es findet unter Beteiligung des Auftraggebers eine bewusste Risikoabwägung statt, wofür es Kenntnisse sowohl des Systems, als auch seiner Anwendung und Verwaltung bedarf.

20.C.2 Es findet eine Differenzierung in Betrachtungsbereich und Tiefe des Tests statt, die von den eingegangenen Risiken und - falls vorhanden - von den Akzeptanzkriterien abhängig ist: Nicht alle Zwischenprodukte und Qualitätsmerkmale werden gleich intensiv getestet.

20.C.3 Entsprechend der gewünschten Testtiefe werden eine oder mehrere formale oder nicht formale Testspezifikationstechniken verwendet.

20.C.4 Auch bei Regressionstests findet eine (einfache aber fundierte) Strategiebestimmung statt, bei der zwischen den Varianten „nur neue Komponenten testen“ und „vollständig neu testen“ gewählt wird.

20.C.5 Die Strategie wird aufgestellt und danach auch ausgeführt. Dass die Durchführung der Tests gemäß Strategie stattfindet, wird überwacht. Falls erforderlich, erfolgt eine Anpassung.

Abhängigkeiten

- Keine.

Optimierungsvorschläge

Siehe die Anweisungen für den Kernbereich „Teststrategie“, Ebene A, Teststrategie für einzelne High-Level Tests.

Es ist ohne Zweifel gut, wenn Entwickler ihre eigene Arbeit testen. Berücksichtigen Sie dabei jedoch die jeweiligen Vor- und Nachteile. Der Vorteil ist, dass die Person ihre eigene Arbeit gut kennt und dadurch sehr schnell passende Testfälle bestimmen und durchführen kann. Außerdem kann sie bei festgestellten Problemen die Ursache schnell finden und Abhilfe schaffen. Der Nachteil dabei ist, dass die „blind spots“ des

Programmierers nicht von ihm selbst entdeckt werden. Erwägen Sie daher die Möglichkeit, Entwickler gegenseitig ihre Arbeiten testen zu lassen (beispielsweise stichprobenartig oder in Integrationstests).

21 INTEGRATIONSTESTS

Beschreibung der Integration

Fehlerhafte Interaktion zwischen (Hardware- oder Software-)Komponenten, Modulen, Subsystemen oder Systemen resultiert oftmals aus Fehlinterpretationen von (in vielen Fällen) nicht formal beschriebenen Anforderungen. Abweichungen, die sich auf diese inkorrekten Interaktionen beziehen, müssen durch Integrationstests entdeckt werden. Diese Tests müssen die korrekte Funktion von zwei oder mehreren zusammenarbeitenden Testobjekten validieren.

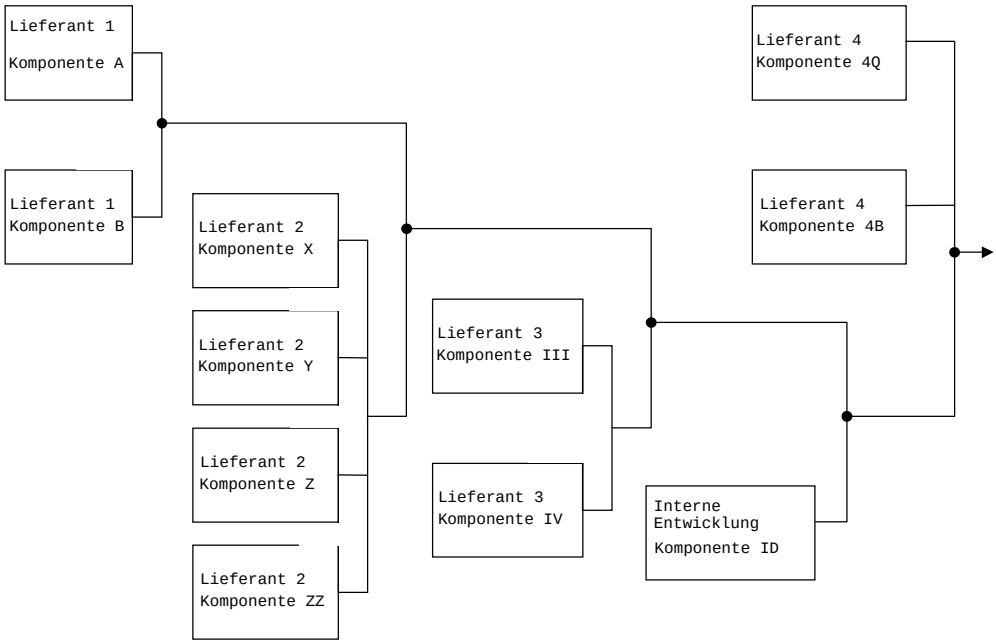
Der Gegenstand dieses Kernbereichs sind die Integration und die Integrationstests eines identifizierbaren Endprodukts, für das ein Abnahmetest definiert ist oder durchgeführt wird. Ein Endprodukt kann eine Komponente, ein Teilsystem oder gar ein komplettes Fahrzeug sein.

Für softwareabhängige Lösungen in Autos beginnt dies mit Softwarekomponenten, die in ein Softwareteilsystem integriert werden müssen. Dieses Softwareteilsystem wird, wo notwendig, zusammen mit anderen Teilsystemen mit einem Mikrocontroller integriert. Der Mikrocontroller wird, falls nötig, zusammen mit anderen Mikrocontrollern zu einem ECU (electronic control unit) integriert. Die ECU wird zusammen mit anderen ECUs zu einem Fahrzeugteilsystem integriert (z. B. Antriebssystem oder Chassis System) und dann werden schließlich diese Steuergeräte in ein komplettes Auto integriert.

Jeder Integrationsschritt muss organisiert, geplant und getestet werden. Bei fast jedem Integrationsschritt ist mehr als eine (interne oder externe) Partei involviert.

Die üblicherweise angewandte Methode der schrittweisen Auslieferung der Funktionalität (sogenannte A-Muster, B-Muster etc.) zwingt den Auftraggeber, die Führungsrolle bei der Entscheidung zu übernehmen, welche Funktionalität in welchem Muster verfügbar sein soll. Eine gewisse Minimalfunktionalität je Systemteil ist notwendig, um dieses Systemteil mit anderen Systemkomponenten integrieren zu können.

Die folgende Abbildung zeigt die typische Abfolge beim Integrationstest:



Integrationstests haben folgende Zielsetzungen:

- Zu Verifizieren, dass das Zusammenstellen der Komponenten (Hardware und/oder Software) nicht zu einer Verschlechterung der Funktionalität der zusammengestellten Komponenten führt.
- Zu Verifizieren, dass die Schnittstelle und die Kommunikation zwischen den zusammengestellten Komponenten (Hardware und/oder Software) der Spezifikation entsprechen.
- Zu Verifizieren, dass die Zusammenstellung als Ganzes stabil und zuverlässig ist.
- Zu Validieren, dass die Zusammenstellung die gewünschte Funktionalität aufweist.

21.A Integration ist als separater und geplanter Prozess identifiziert

Beschreibung

Die Organisation ist sich bewusst, dass alle separaten (Hardware- und/oder Software-) Komponenten in ein finales Produkt zu integrieren sind. Als Ergebnis wurde mindestens ein Integrationstest im Rahmen der Projektplanung identifiziert, einschließlich der Beginn- und Endtermin. Die Einhaltung von Beginn- und Endterminen wird durch das Einbeziehen der Einkaufsabteilung wahrscheinlicher.

Kontrollpunkte

21.A.1 Eine Person ist verantwortlich für die Integration der einzelnen Komponenten in ein Gesamtsystem, einschließlich Integrationstests. (Rolle des Integrators)

21.A.2 Die Reihenfolge der Auslieferung von dem (den) Lieferanten zum Testen wurde im Voraus festgelegt und dokumentiert.

21.A.3 Die Integrations- und Testabfolge sowohl für die Hardware als auch für die Software (die sogenannte Integrationsstrategie) wurde im Voraus auf Basis der (Software-) Architektur und des Lieferplans der Komponenten definiert und dokumentiert. Diese Planung beinhaltet:

- Aktivitäten
- Abhängigkeiten
- Meilensteine

21.A.4 Die Übergabe an und vom Testteam findet nach einer Standardprozedur statt. Der Gegenstand der Übergabe ist bekannt (und in Form eines Übergabeberichts reproduzierbar beschrieben):
Welche Komponenten und Versionen des Testobjekts, welche Version der Testbasis, (nicht) gelöste Abweichungen, Konfiguration

21.A.5 Im Falle von Abweichungen vom Plan (z. B: Spätlieferung der Komponenten) werden Anpassungen vorgenommen. Die Abweichungen werden begründet.

Abhängigkeiten

- Testfunktionen und Training, Ebene A, Testmanager, Integrator und Tester
- Testprozessmanagement, Ebene B, Planung, Durchführung, Überwachung und Anpassung

Optimierungsvorschläge

Beginnen Sie damit, frühzeitig einen Integrator zu benennen, der verantwortlich für die Integration der Komponenten und das Testen ist. Sorgen Sie dafür, dass diese Person eine formale Rolle im Projekt hat.

Der Integrator gibt mehr Fokus in den Integrationsprozess, kann das Bewusstsein steigern und den Reifegradprozess initiieren

21.B Teststrategie für die Integration

Beschreibung

Die Kombination der Komponenten und die Reihenfolge der Tests werden von den erkannten Risiken beeinflusst. Trotz korrekten Modultests kann eine Integration dennoch fehlschlagen.

Kontrollpunkte

21.B.1 Die Sequenz der Lieferungen und die Eingangskriterien für die zu liefernden Komponenten werden zwischen Integrator und Komponentenlieferant(en) abgestimmt und vom Integrator koordiniert und dokumentiert.

21.B.2 Die Anzahl und die Reihenfolge der Integrationsschritte basieren auf einer bewussten Überlegung der erwarteten Risiken des Produkts oder der Produktkomponenten, der technischen Restriktionen und der Auswirkungen der Veränderungen.

21.B.3 Eingangskriterien für die Komponenten und Ausgangskriterien für die Integrationstests wurden definiert, dokumentiert und angewendet. Diese Kriterien werden vorzugsweise in Form von erreichter Testabdeckung und/oder von ungelösten Abweichungen definiert.

21.B.4 Es gibt eine Differenzierung in der Testtiefe, abhängig von den Risiken und (sofern vorhanden) von den Eingangskriterien und Ausgangskriterien:
Nicht alle Komponenten, Varianten und Versionen werden gleichermaßen gründlich getestet und nicht alle Qualitätskriterien werden (gleichermaßen gründlich) getestet.

21.B.5 Auch für Wiederholungstests findet eine (einfache) Strategiebestimmung statt, in der eine überlegte Entscheidung zwischen „nur Testen der Lösungen" und „vollständige Testwiederholung" getroffen wird.

21.B.6 Abweichungen von der koordinierenden Strategie werden berichtet, woraufhin eine begründete Anpassung der koordinierenden Strategie auf Basis der identifizierten Risiken erfolgt.

21.B.7 Vereinbarungen für die Unterstützung durch den Komponentenlieferanten werden getroffen. Diese Vereinbarungen beinhalten: Fehlerbehebung, Behebung von testblockierenden Abweichungen, Kommunikationsstrukturen (z. B. regelmäßige Integrationsmeetings) und Eskalationsprozedur.

Abhängigkeiten

- Phasenmodell, Ebene B, Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss
- Zeitpunkt der Beteiligung, Ebene B, Beginn der Testbasis

Optimierungsvorschläge

Der Integrator sollte Zugang zu allen relevanten Zwischenprodukten (Design, Quellcode (sofern er als Teil der Lieferung definiert ist, und alle Testprodukte) des Komponentenlieferanten haben.

Bei der Definition der Teststrategie sollte man folgenden Punkten bei der Ermittlung des Risikos besondere Aufmerksamkeit widmen:

- Die Funktionalität der (Hardware oder Software) Komponenten,
- Die Komplexität von (Hardware- oder Software-)Komponenten und Ihre Schnittstelle,
- Die bereitgestellten Qualitätsnachweise für (Hardware- oder Software-) Komponenten (über Testfälle, Testberichte, Zertifikate, etablierte Anwenderbasis etc,)
- Die Transparenz der (Hardware- oder Software-)Komponenten. Transparenz sagt aus, wie leicht man herausfinden kann, wie das Teil arbeitet und sie ist wichtig für die Fehleranalyse. Transparenz lässt sich z. B.: erreichen durch Originalspezifikationen, die vom Hersteller der (Hardware- oder Software-) Komponenten bereitgestellt werden, indem man den Quellcode zur Verfügung stellt durch so genannte „Testfenster" in der Software, oder durch die Anzeige von Zwischenergebnissen.
- Die Häufigkeit der Verwendung der (Hardware- oder Software-) Komponenten,
- Der mögliche Schaden, wenn ein spezifischer Teil der Software ausfällt,
- Die Verfügbarkeit von (Test-)Ressourcen (sowohl Personal als auch Testinfrastruktur).

21.C Standardisierter Ansatz für die Integration

Beschreibung

Der Ansatz für die Integration wird in Vorgehensweisen dokumentiert. Diese Prozeduren sind jedoch nicht statisch, sondern sie werden auch aufgrund praktischer Empfehlungen angepasst und gepflegt. Hieraus resultiert ein Standardvorgehen für Integrationstests innerhalb der Firma (anwendbar für alle Projekte) und der Integrator hat Einfluss auf die Lieferungsreihenfolge der (Hardware- und Software-) Komponenten.

Kontrollpunkte

21.C.1 Die Prozeduren sind in einem generischen Ansatz für die Organisation definiert.

21.C.2 Jedes Projekt arbeitet nach diesem generischen Ansatz.

21.C.3 Abweichungen von Standardvorgehen werden hinreichend begründet und dokumentiert.

Abhängigkeiten

- Keine

Optimierungsvorschläge

Dokumentieren Sie die Vorgehensweise im Handbuch oder beziehen Sie sich auf Literatur.

In regelmäßigen Intervallen (z.B.: halbjährlich) sollten Sie die Integrationsstrategie, die Integrationsvorgehensweise und die verfolgte Strategie bewerten und bei Bedarf aktualisieren.

Legen Sie die Verantwortung, die Prozeduren zu formulieren, zu überwachen und zu pflegen in die Hände einer Linienabteilung. Diese Abteilung sollte über ein hinreichendes Maß an Testexpertise verfügen, um diese Rolle auszufüllen.

Kontrollieren und steuern Sie die Implementierung des Innovationsleitungskreises (Planung + Initialisierung der Ressourcen).

22 TESTREIFEMATRIX

Kernbereich														
1	**Teststrategie**		A					B				C		D	
2	**Einsatz des Phasenmodells**		A			B									
3	**Zeitpunkt der Beteiligung**			A				B				C		D	
4	**Kostenvoranschlag und Planung**				A							B			
5	**Testspezifikations-techniken**		A		B						C				
6	**Statische Testtechniken**					A		B							
7	**Metriken**						A			B			C		D
8	**Testautomatisierung**				A				B			C			
9	**Testumgebung**				A				B						C
10	**Büro- und Laborausstattung**				A										
11	**Engagement und Motivation**		A				B						C		
12	**Testfunktionen und Training**				A			B			C				
13	**Reichweite der Methodik**					A		B				C			D
14	**Kommunikation**			A		B							C		
15	**Berichterstattung**		A			B		C					D		
16	**Dokumentation der Abweichungen**		A				B		C						
17	**Testware Management**			A			B				C				
18	**Testprozessmanagement**		A		B								C		
19	**Prüfen**			A				B			C				
20	**Low-Level Tests**					A		B		C					
21	**Integrationstests**				A			B				C			

23 ÜBERBLICK ÜBER DIE ABHÄNGIGKEITEN

Bei jeder Ebene sind in Klammern die Abhängigkeiten von anderen Ebenen angegeben. Beim Kernbereich „Teststrategie", Ebene A, stehen beispielsweise zwei Abhängigkeiten: 5a und 11a. Die Nummer weist auf den Kernbereich hin und der Buchstabe auf die Ebene. In diesem Beispiel ist die Ebene A der Teststrategie also von den Testspezifikationstechniken, Ebene A, und Engagement und Motivation, Ebene A abhängig.

	Kernbereich	**A**	**B**	**C**	**D**
1	–Teststrategie	–Strategie für einzelne High-Level Tests (5a, 11a)	–Kombinierte Strategie für High-Level Tests (2a, 5b, 11b, 14b, 18b) und abhängig vom Bereich des Integrationstests (21b)	–Kombinierte Strategie für High-Level Tests und Low-Level Tests oder Prüfungsstufen (3c, 19c) oder (20c) oder (21b) im Falle einer Low-Level Integration	–Strategie für alle Test- und Prüfungsstufen (3c, 19c, 20c, 21b)
2	–Einsatz des Phasenmodells	–Planung, Spezifikation, Durchführung (11a)	–Planung, Vorbereitung, Spezifikation, Durchführung und Abschluss (6a, 17a)	–	–
3	–Zeitpunkt der Beteiligung	–Fertigstellung der Testbasis (2a)	–Aufstellen der Testbasis (2b)	–Aufstellen der Anforderungen	–Beginn des Projekts (11c)
4	–Kostenvoranschlag und Planung	–Fundierte Kostenvoranschlag und Planung (2a)	–Statistisch fundierter Kostenvoranschlag und Planung (7b, 15b)	–	–
5	–Testspezifikations-techniken	–Nicht formale Techniken	–Formale Techniken (12a, 17a)	–Mathematische Methoden	–
6	–Statische Testtechniken	–Inspektion der Testbasis	–Checklisten	–	–
7	–Metriken	–Projektmetriken	–Projektmetriken (über	–Systemmetriken	–Organisationsmetrik

		(über Produkt) (11b, 15b, 16a, 18b)	Prozess) (15c, 16b)	(13c, 14c, 18c)	en (>1System)
8	–Testautomatisierung	–Einsatz von Testtools	–Beherrschung der Testautomatisierung (5a oder 5b, 12a)	–Optimale Testautomatisierung	–
9	–Testumgebung	–Kontrollierte Testumgebung (12a)	–Testen in der geeignetsten Umgebung (1b)	–„Umgebung auf Abruf"	–
10	–Testarbeitsplatz	–Adäquate und rechtzeitige Einrichtung der Testarbeitsplätze	–	–	–
11	–Engagement und Motivation	–Zuweisung von Budget und Zeit	–Testen in Projektorganisation integriert (2a, 15b, 16a, 18b)	–Testengineering (1c, 3c, 8b, 15c)	–
12	–Testfunktionen und Ausbildung	–Testmanager und Tester	–(Formale) Unterstützung, methodische technische und funktionale Management	–Formale interne Qualitätssicherung (13a)	–
13	–Reichweite der Methodik	–Projektspezifisch (2b, 5b, 16a, 17a, 18b)	–Projektspezifisch mit externen Bereich (1b, 14b, 15b, 16b, 17b)	–Organisations-generisch	–Organisations-optimierend, F&E Aktivitäten (11b, 18c)
14	–Kommunikation	–Interne Testkommunikation	–Projektkommunikation (Analyseforum, Änderungsüber-wachung) (2a, 15b, 16a)	–Kommunikation über die Qualität der Testprozesse auf Organisations-ebene (13c)	–

15	–Berichterstattung	–Aufdeckung der Abweichungen	–Abweichungen, einschließlich Prioritätenzuweisung und Berichterstattung über Zeitaufwand und Testfortschritt(2a, 16a, 18b)	–Risiken und Empfehlungen anhand von Metriken (1a, 5b, 7a, 16b)	–Empfehlungen mit SPI ('Software Process Improvement') Abschnitt (1c, 11c)
16	–Dokumentation der Abweichung	–Interne Dokumentation der Abweichungen	–Umfangreiche Dokumentation der Abweichungen mit flexiblen Berichterstattungs-möglichkeiten	–Dokumentation der Abweichungen wird im gesamten Projekt eingesetzt	–
17	–Testware Management	–Internes Testware Management	–Externes Management von Testbasis und Testobjekt	–Übertragbare Testware (5b)	–
18	–Testprozess Management	–Planung und Durchführung	–Planung, Durchführung, Überwachung und Anpassung	–Überwachung und Anpassung in der Organisation (13c)	–
19	–Prüfen	–Überprüfungs-techniken	–Überprüfungsstrategie	–Überprüfungs-strategie	–
20	–Low-Level Tests	–Phasenmodell für Low-Level Tests (Planung, Spezifikation und Durchführung)	–(White-Box) Techniken	–Low-Level Teststrategie	–
21	–Integrationstests	–Identifizierte Integration als separaten und geplanten Prozess (12a, 18b)	–Teststrategie für Integration (2b, 3b)	–Strategische Annäherung an Integration	–

24 GLOSSAR

Anforderungsabdeckung	Der Deckungsgrad wird häufig als % Zahl wiedergegeben.
Aktualisierbarkeit	Die Mühelosigkeit, mit der das System an die neuen Wünsche der Anwender oder die geänderte externe Umgebung angepasst oder eingesetzt werden kann, um Fehler zu korrigieren.
Akzeptanzkriterien	Kriterien, die definieren, wann welche der Bedingungen erfüllt sein müssen, damit das Objekt zum Test freigegeben wird.
Akzeptanztest	Ein Test, der von einem Benutzer und/ oder dem Systemmanager in einer Umgebung durchgeführt wird, die den Praxisanforderungen nahe kommt, um zu zeigen, dass das entwickelte System die Funktions- und Qualitätsanforderungen erfüllt.
Anforderungen	Eine Bedingung, die entweder durch das System, einer Komponente oder einer Funktion getroffen werden muss, damit ein Vertrag, ein Standard oder ein anderes formales Dokument erfüllt wird.
Ausfall	Eine Abweichung im System vom erwarteten Ergebnis (in der Lieferung oder Service).
Ausgangssituation	Der Status zu Beginn einer Testphase. Hier werden gewöhnlich die Ausgangsdaten und die Variablen beschrieben.
Auswertung	Das Kontrollieren und Durchführen von Reviews an verschiedenen Zwischenprodukten und/oder in der Software bzw. im Systementwicklungsprozess.
Fahrzeug Prototyp	Das Fahrzeug, das die Tests ausführt.
Beauftragter	Die Person, die für die Zuweisung verantwortlich ist.
Bekannte Fehler	Fehler die gefunden, aber bis jetzt nicht gefixt worden sind.
Big-bang Integration	Es handelt sich um eine Integration aller Komponenten in einem Schritt, bei der keine zusätzliche Integration von Bestandskomponenten durchgeführt wird.

Black-Box Testtechnik	Eine Kategorie von Testtechniken, bei denen Testfälle, ohne Kenntnis des internen Konzepts eines Objekts, von den externen sichtbaren Eigenschaften eines Objekts abgeleitet werden.
Bottom-up Testen	Hierbei handelt es sich um einen Test, bei dem die niedrigsten Komponenten zuerst getestet werden. Anschließend darauf aufbauend die höher liegenden Komponenten, usw. Der Prozess wird solange wiederholt, bis man an der Spitze der Testhierarchie angekommen ist.
Checkliste	Eine Checkliste ist eine Liste von Fragen, die nur mit 'Ja' und 'Nein' beantwortet werden können.
Code Checker	Eine übergeordnete Bezeichnung für verschiedene Tools, um Codes auf mögliche Fehler (z.B. Speicherfehler, schlechte Freigabe) hin zu analysieren.
Code review	Ein Meeting bei dem der Software Code dem Projektmanager, den Benutzern und dem Kunden oder anderen interessierten Personen im Testbereich präsentiert wird.
COTS	„Commercial off the Shelf" bezeichnet kommerzielle Systemkomponenten oder auch Software.
Coverage	Das Verhältnis zwischen dem was getestet werden kann und dem was getestet wird.
Code Coverage	Eine Analysemethode, die festlegt, welche Komponenten einer Software durch Testfälle getestet (abgedeckt) werden können und welche nicht getestet (abgedeckt) werden können.
Design (Planung)	Ein Prozess, der die Architektur, Komponenten, Schnittstellen und andere Eigenschaften eines Systems oder einer Komponente definiert.
Dynamisches Testen	Testen auf der Grundlage gezielter Testfälle durch Ausführen des Testobjekts und/oder Programmen.
Ebene	Im TPI Modell sind die Kernbereiche in verschiedenen Ebenen unterteilt.
ECU	„Electronic Control Unit" bezeichnet die Software eines Steuergeräts.
Eingabewerte	Der klassische Fall einer Eingabe.

Eingang (Input)	Ein Signal oder eine Variable, die durch das System oder der Komponente verarbeitet wird.
Eingangsprüfung	Eine Prüfung, die feststellt, ob das Testobjekt komplett geliefert wurde und bestimmt, wann die nächste Testphase gestartet werden kann.
Eingebettete Software	Software in einem eingebetteten System, um als Aufgabe die spezifische Hardware eines Systems zu kontrollieren.
Eingebettete Systeme	Ein System, das mit der realen Welt interagiert durch Benutzung von Sensoren und Aktionen.
Einheit	Eine Softwarekomponente, die sich nicht weiter aufteilen lässt.
Emulator	Eine Vorrichtung, ein Programm oder ein System, das die gleichen Eingaben akzeptiert und die gleichen Ausgaben wie das vorgegebene System produziert.
Ergebnis	Siehe aktuelles Ergebnis oder erwartetes Ergebnis.
Erwartete Ergebnis	Das erwartete Verhalten durch Spezifikation eines Objektes mittels eines spezifischen Tests.
Fehler	Siehe Störung.
FMEA	„Failure Mode and Effect Analysis" bezeichnet eine analytische Methode, um potentielle Schwachstellen zu finden
Funktionale Spezifikation	Ein Dokument, das die Funktionen spezifiziert, die ein System oder eine Komponente erbringen muss.
Grenzwertanalyse	Das Testprinzip basiert auf der Tatsache, dass ein Test rund um den Grenzwert eine größere Fehlerfindungschance bietet.
Hardware	Hardware ist ein Sammelbegriff für alle Baugruppen und Peripheriegeräte eines Computers.
High-Level Tests	Diese Teststufen werden zum Testen abgeschlossener, kompletter Produkte eingesetzt.
HiL	„Hardware in the Loop" ist eine Testebene, bei der die Hardware eines Systems über spezielle Schnittstellen in einer simulierten Umgebung getestet wird.
Inspektion	Ein Gruppenqualitätsverbesserungsprozess für schriftliches Material. Es beinhaltet zwei Aspekte, zum einem die Verbesserung des Produktes (das eigene

	Dokument) und zum anderen die Verbesserung des Reifeprozesses.
Integration	Ein Prozess von Kombinationen verschiedener Komponenten zu größeren Einheiten.
Integrationsstrategie	Eine Entscheidung, wie verschiedene Komponenten in ein komplettes System integriert werden sollen (Integrationsstrategien sind: big-bang, top-down und bottom-up Integration).
Integrationstest	Ein Test, der nachweisen soll, dass das Zusammenspiel von Programmkomponenten den in den technischen Spezifikationen gestellten Anforderungen entspricht.
Kodierung	Die Transformation von Logik und Daten von Designspezifikation (Designbeschreibung) in die Programmsprache.
Komponenten	Hierbei kann es sich um Hard- oder Software handeln, die für die eine separate Spezifikation vorhanden ist. Eine Komponente kann wiederum in einzelne Komponenten unterteilt werden.
Komponententest	Der Test von einzelnen Komponenten oder zusammenhängenden Komponenten als Gruppe.
Kontrollpunkte	Zur objektiven Bestimmung der Ebene verfügt das TPI Model über Messinstrumente, den so genannten Kontrollpunkten.
Lebenszyklus	Ein Lebenszyklus strukturiert den Prozess, indem er ihn in Phasen einteilt und beschreibt.
Low-Level Tests	Der Prozess einzelne Komponenten allein oder in Kombination zu testen.
Mastertestplan	Ein Testplan, in dem die unterschiedlichen Test- und Prüfungsstufen abgestimmt werden.
Metriken	Die objektiven, quantitativen und gesammelten Anhaltspunkte, um die Eigenschaften eines Produktes oder eines Prozesses zu überwachen.
MiL	„Model in the Loop" ist eine Testebene, bei dem das Simulationsmodel eines Systems in einer dynamischen Testumgebung getestet wird.
Model Checker	Ein Tool, das ein mathematisches Model in einem System entgegen spezifizierten

	Korrekturanforderungen analysiert.
Output	Ein Signal oder eine Variable, die als Ausgabe einer Komponente erscheint.
Output Werte	Eine Gruppe von Ausgaben.
Prüfen	Überprüfung und Inspektion der verschiedenen Zwischenprodukte im Entwicklungszyklus.
Qualität (ISO 8402)	Qualität ist die Gesamtheit von Merkmalen eines Produktes oder einer Dienstleistung bezüglich ihrer Eignung, festgelegte und vorausgesetzte Erfordernisse zu erfüllen[ISO 8402, 1994].
Qualitätsmerkmale	Die Eigenschaft eines Systems. Einige Beispiele sind: Richtigkeit, Sicherheit, Leistung und Benutzerfreundlichkeit.
Regressionstest	Mit Regression wird das Phänomen bezeichnet, dass sich die Qualität eines Systems infolge individueller Anpassungen verringern kann. Ein Regressionstest zielt darauf ab zu kontrollieren, ob alle Elemente eines Systems nach einer Änderung noch korrekt funktionieren
Risiko	Generell definiert als den maximalen Verlust multipliziert mit der Möglichkeit eines Fehlerauftretens.
Kernbereich	Werden in ihrer Betrachtung durch das TPI Model aufgefangen.
Schnelles Prototyping	Eine Testtechnik, wobei das simuliert eingebettete System mit Verbindung zur realen Welt getestet wird.
Aufnahme/ Wiedergabe-Tool	Ein Testtool, das die Testeingaben aufzeichnet, sobald diese im Test eingegeben werden. Der Speicher in dem Testtool kann benutzt werden, um den Test mit genau den gleichen Eingabedaten zu wiederholen.
SiL	„Software in the Loop" ist eine Testebene, wobei die reale Software in einer simulierten Umgebung getestet wird.
Simulation	Das ist eine Vorgehensweise zur Analyse von dynamischen Systemen.
Simulator	Ein Computerprogramm oder ein System, das genauso läuft wie ein Echtzeitsystem und mit kontrollierter Eingabe gesteuert wird.
Software	Computerprogramme, bzw. Prozesse, die zur Verarbeitung auf einem Computersystem benötigt

	werden.
Software unter Test	Die Software, die in der Testphase für einen Test benutzt wird.
Spezifikation	Eine Beschreibung, wie die Komponenten zu funktionieren haben und deren Verhalten bei bestimmten Eingabewerten.
Anforderungsspezifikation	Ein Dokument, das die Anforderungen spezifiziert.
Statische Analyse	Testen durch Überprüfung und Untersuchung von Produkten, ohne dass Programm(teil)e ausgeführt werden..
Statischer Test	Der Test eines Objektes ohne Ausführung auf einem Computer
Störung (Defekt)	Die Manifestation eines Fehlers in der Software, meistens durch eine Fehlermeldung sichtbar gemacht.
Stub	Ein simulierter Teil eines Softwaremoduls, um das Testen für bestimmte Funktionen zu unterstützen.
System	Eine Kollektion von Komponenten, die spezielle Funktionen übernehmen.
Systemkomponenten	Teil eines kompletten Systems
Systemtest	Ein Test, der nachweisen soll, dass das entwickelte System oder Teile davon den in den Spezifikationen festgelegten Anforderungen entspricht.
Test- und Prüfungskoordinator (TPK)	Dieser ist verantwortlich für die Identifikation und Definition der Testobjekte und Testpakete. Weiterhin unterstützt er die technische Leitung und generiert den Testbericht.
Teststufe	Eine Reihe von Gruppenaktivitäten, welche organisiert und verwaltet werden, entweder als High-Level oder als Low-Level Tests.
Testaktion	Teil des Testfalles, der die Aktionen beschreibt.
Testautomation	Der Gebrauch von Software um einzelne Tests zu automatisieren.
Testware Review	Die ausführliche Auswertung der zu testenden Grundlage.

Testbasis	Alle Dokumente, die an einem System gestellt werden, bzw. auf welcher der Test beruht.
Testendekriterium	Ein Kriterium, das festlegt, wann der geplante Test fertig ist.
Testdesigntechnik	Eine standardisierte Methode von verschiedenen Tests (z.B. elementarer Vergleichstest oder Strukturtest)
Testdurchführung	Ausführung der Testfälle bzw. Testszenarien (Aktivität im Testprozess)
Testen	Der Prozess des Planens, der Vorbereitung und des Messens, mit dem Ziel, die Merkmale eines Systems festzustellen und Unterschiede nachzuweisen.
Testfall	Eine Beschreibung eines auszuführenden Tests.
Testinfrastruktur	Die Umgebung, in der die Tests durchgeführt werden.
Testmesstechnik	Eine Methode zeigt das Maß der Testabdeckung (z.B. die sich aus den Testfällen und der Codeabdeckung ergebenen Anforderungen.
Testobjekt	Das zu testende System
Testorganisation	Eine Testorganisation hat die Aufgabe, adäquate Verhältnisse zwischen Testfunktionen, Testeinrichtungen und Testaktivitäten zu schaffen, um rechtzeitig eine gute Qualitätsempfehlung formulieren zu können.
Testplan	Dort wird das allgemeine Konzept und die strategischen Entscheidungen im Zusammenhang mit dem auszuführenden Test festgelegt
Testablauf	Eine Planung der auszuführenden Testskripte; die Testskripte stehen im Testablauf in einem logischen Zusammenhang und sind in der auszuführenden Reihenfolge angegeben.
Testprozess	Der Prozess zur Organisation, dem Verwalten und das Ausführen von Tests.
Testreifematrix	Ein strukturierter Überblick über die TPI Analyse.
Testsatz	Eine Sammlung von Testfällen.
Testskript	Eine Beschreibung, wie das Testen durchgeführt wird. Es beinhaltet Testdurchführungen und Kontrollen

	bezogen auf konkrete Testfälle
Testspezifikation	Eine Beschreibung der Art und Weise, wie logische Testfälle ausgewählt sind.
Teststrategie	Die Aufteilung von Testaufwand und Überdeckungsgrad über die zu testenden Komponenten oder Aspekte des Testobjektes, damit Fehler so früh als möglich gefunden werden, um Kosten zu vermeiden
Testteam	Eine Gruppe von Mitarbeitern, die Unter der Leitung eines Testmanagers die Testaktivitäten durchführt.
Testtechnik	Sammlung von Aktionen, um ein Testprodukt auf allgemeine Weise herzustellen..
Testtool	Ein automatisches Hilfsmittel, zur Unterstützung für eine oder mehrer Testaktivität(en)
Testumgebung	Das Zusammenspiel von Komponenten, wie Hardware, Software für den Aufbau und die Nutzung von Daten, in denen ein Test ausgeführt wird.
Testware	Alle Testdokumente, die während des Testprozesses hergestellt werden.
Treiber	Ein Gerüst oder eine spezielle Implementierung eines Software Moduls, um Testkomponenten zu unterstützen, die durch den Treiber ausgelöst werden.
Übereinstimmungskriterien	Ein Kriterium, um zu beurteilen, ob die Durchführung des Bestandteiles in Übereinstimmung mit dem Bestandteil einer Spezifikation passt.
Überprüfbarkeit	Die Problemlosigkeit, mit der die Richtigkeit und Vollständigkeit von Informationen überprüft werden.
Verhalten	Die Kombination aus eingegebenen Werten als Vorbedingung für ein funktionierendes System. Die komplette Spezifikation würde normalerweise ein oder mehrere Verhalten aufzeigen.
Walkthrough	Eine manuelle, informale Prüfmethode, um Fehler, Defekte, Unklarheiten und Probleme in schriftlichen Dokumenten zu identifizieren. Der Autor präsentiert das Dokument in einer Sitzung den Gutachtern.
White-Box Testtechnik	Eine Kategorie von Testtechniken, bei denen Testfälle mit Kenntnissen der internen Struktur des Objekts von den internen Eigenschaften eines Objekts abgeleitet werden.

25 LITERATUR

Beizer B. (1990). Software Testing Techniques. International Thomson Computer Press

Binder R.V. (2000). Testing object-oriented systems, Models, Patterns, and Tools. Addison Wesley

Boehm, B.W. (1981). Software Engineering Economics, Prentice Hall.

Broekman E and Notenboom E (2003). Testing Embedded Software. Addison-Wesley

Hetzel B. (1993). Making Software Measurement Work: Building an Effective Measurement Program (Qed Software Evalutation)

Kaner C., Falk J. Nguyen H. Q. (1999). Testing Computer Software, 2nd Edition. Wiley

Koomen T., Pol M. (1999). Test Process Improvement – A practical step-by-step guide structured testing. Addison Wesley

Pulford K., Kunntzmann-Combelles A. and Shirlaw S. (1996). The ami handbook, A quantitative approach to software management. Addison Wesley

Rushby J. (1995). Formal methods and their Role in the Certification of Critical Systems. Computer Science Laboratory SRI International